Data Driven Seizure and Ictal-Interictal Continuum Pattern Detection in Electroencephalography

Building ML solutions for Detection of Seizure, Rhythmic and Periodic patterns in the ICU setting

by

Joris van Eekeren

Student number:	8865736	
Project duration:	January 1, 2024 – July 23, 2024	
Thesis committee:	Chris Janssen,	Utrecht University, 1st reader
	Sam Chota,	Utrecht University, 2nd reader
Daily supervisors:	Brandon Westover,	Harvard Medical School, Principal Investigator
	Jin Jing,	Harvard Medical School, Daily Supervisor

Utrecht University

Preface

The past seven months I had the pleasure of being a part of the Clinical Data Animation Centre, headed by Dr. Brandon Westover. I would like to thank Brandon and Jin Jing for giving me the chance to come overseas in order to work together on this interdisciplinary project, investing time in me by being my daily supervisor, and providing me with knowledge about Neurology and EEG. Getting lectures about EEG patterns and seeing EEG recordings live from patients were extremely interesting experiences. The experience of conducting research in such an exciting field with real patient data while living in the US is one I enjoyed greatly and will carry with me for the rest of my life. I would also like to thank Chris Janssen for giving excellent advice and guiding the academic process from back in the Netherlands. The sharp feedback and writing tips helped me enormously in making this thesis. Last but not least, I would like to give special thanks to Niels Turley, our research assistant, for helping me install many hardware components and overcome computational problems.

Joris van Eekeren
Boston, July 2024

Abstract

This research aims to take steps in developing Seizure and Ictal-Interictal Injury Continuum (IIIC) detection models on long and noisy EEG recordings from critically ill ICU patients because automated detection can significantly help neurologists and patients by diagnosis of serious brain diseases like tumors and epilepsy. This work is a continuation of the work by Jin Jing et al. who developed a DenseNet Convolutional Neural Network called Sparcnet for the same purpose. Sparcnet is capable of classifying seizures, lateralized/generalized periodic discharges and rhythmic delta activity (LPD, GPD, LRDA, GRDA), as well as everything else defined as ”Other”. However, this deep learning model was not trained on longer EEG recordings and had high false positive rates on this type of noisy data. In order to improve the quality of Seizure and IIIC labels for raw 10-second EEG segments, the labels from experts were passed through a Bag of Words algorithm. This algorithm improved labels where power spectral density values were peaking but could not be applied universally to all patients and had to be tuned by an expert. The results suggest that the method has limitations in the form of a bias which results from only using one expert and clustering only on power spectral density. These labels were used in transfer learning experiments with different parameters and the Sparcnet model. This showed that the transfer learning experiments failed to outperform Sparcnet 1. The main indications for this result were that the label quality is still not good enough and that there is a need for more computing power to do proper hyperparameter tuning. A Boosting Cascade and Multinomial Logistic Regression (MLR) were applied as postprocessing steps on the Sparcnet model to provide a less resource-heavy approach to the problem and try to reduce the false positive rate of Sparcnet. The results showed that the Boosting Cascade was able to slightly improve almost all macro evaluation metrics and reduce the false positive rate of classes at the cost of increasing the false positive rate of the ”Other” class. The MLR model was able to reduce the false positive rate for the ”Other” and ”Seizure” classes. Both methods showed to be able to recognize clear artifacts that Sparcnet wrongfully classified as seizures. The results suggested that both methods rely too much on the Sparcnet output, which is not optimal, and that they might not be able to capture the complex nonlinear patterns in the EEG data. A number of recommendations resulted from this work. First, the labels are not ready and need to be improved by having experts do another round of labeling while they validate each other’s labels with the graphical user interface that was developed with the Bag of Words method. Second, in order to exclude hyperparameters as a potential reason for bad performance improvements can be made by implementing adaptive learning rates, k-fold cross-validation, and advanced hyperparameter tuning. Third, these improvements need enough computing power in the form of a High Performance Computing Cluster. Fourth, more research can be done on a data-level approach to solve the imbalance in the dataset. Fifth, more patterns need to be added to the data. The ”Other” class for example needs to be split in a ”Normal” and ”Artifact” class.

Contents

Preface i

Abstract ii

Introduction 1

1 Project and Research Description 3
1.1 Related work . . . 3
1.2 Problem Definition and Research Questions . . . 4
1.2.1 Introduction . . . 4
1.2.2 Problem 1 : Limited High Quality Data Availability . . . 4
1.2.3 Problem 2 : Reduced performance on noisy continuous EEG-data . . . 5
1.2.4 Problem 3 : High False Positive Rate . . . 5
1.3 Outline . . . 7

2 Background 8
2.1 Electroencephalography . . . 8
2.2 IIIC patterns . . . 9
2.3 Convolutional Neural Networks . . . 11
2.4 DenseNet . . . 13

3 Bag of Words and Change Point Detection 15
3.1 Label Quality . . . 15
3.2 Change Point Detection . . . 16
3.3 K-means clustering . . . 18
3.4 Bag of Words . . . 18
3.5 Results . . . 19
3.6 Discussion . . . 20
3.7 Limitations . . . 21

4 Transfer learning with Dense Convolutional Neural Networks 22
4.1 Data . . . 22
4.2 Data pre-processing . . . 22
4.3 Loss function . . . 23
4.4 Computational setup and distributed training . . . 24
4.5 Hyper-parameters and experiments . . . 24
4.6 Evaluation metrics . . . 24
4.7 Results . . . 27
4.7.1 Performance on ROC curves . . . 27
4.7.2 Performance on PR curves . . . 28
4.7.3 Evaluation metrics . . . 29
4.7.4 Confusion matrices . . . 30
4.7.5 Swimmer plots . . . 30
4.8 Discussion . . . 34
4.9 Limitations . . . 34

5 Feature based post processing 36
5.1 Boosting Cascade . . . 36
5.2 Multinomial Logistic Regression . . . 38
5.3 Results . . . 39
5.3.1 Performance on ROC curve . . . 39
5.3.2 Performance on PR curve . . . 39

5.3.3 Macro Evaluation Metrics . . . 40
5.3.4 Confusion Matrices . . . 40
5.3.5 Swimmer plots . . . 40
5.4 Discussion . . . 40
5.5 Limitations . . . 43

6 General Discussion **47**
6.1 Summary of Results . . . 47
6.2 Summary of Limitations . . . 47
6.3 Context to related work . . . 48
6.4 Future work . . . 48

References **50**

A Supplementary Materials **57**
A.1 Bag of Words and Change Point detection parameters . . . 57
A.2 Patient information . . . 59
A.3 DenseNet convolutional neural network architecture . . . 62
A.4 Training losses . . . 65

List of Figures

2.1 Waves and their frequency content [57] . . . 9
2.2 Spectrogram of real seizure [57] . . . 10
2.3 IIIC patterns . . . 11
2.4 Rhythmic Delta Activity vs Periodic Discharges [24] . . . 12
2.5 Convolutional operation [55] . . . 12
2.6 Denseblock [25] . . . 14
2.7 DenseNet architecture [25] . . . 14

3.1 Color code of classes for swimmer plots . . . 16
3.2 glad001 and goodSZ001: Expert labels, Average power (dB) and spectrograms for brain regions (LL, RL, LP, RP) . . . 17
3.3 glad022 and goodSZ001: Expert labels, Average power (dB) and spectrograms for brain regions (LL, RL, LP, RP) . . . 17
3.4 glad001 and goodSZ001: Bag of Words clusters (BOW), new labels (SZ, C), Average power (dB), for brain regions (LL, RL, LP, RP) . . . 19
3.5 glad022 and goodSZ001: Bag of Words clusters (BOW), new labels (SZ, C), Average power (dB), for brain regions (LL, RL, LP, RP) . . . 20
3.6 Bag of Words and Change Point Detection on glad003 under different settings for number of clusters (50 and 30) . . . 20

4.1 Confusion Matrix . . . 25
4.2 MCC and F1 compared to accuracy over thresholds . . . 27
4.3 ROC curves for transfer learning experiments . . . 28
4.4 PR curves for transfer learning experiments . . . 29
4.5 Confusion Matrix Sparcnet 1 . . . 31
4.6 Confusion Matrix Exp 4 . . . 32
4.7 glad004 and glad017 transfer learning swimmer plots and spectrograms . . . 32
4.8 Raw EEG glad017 . . . 33

5.1 Training steps of the cascade . . . 38
5.2 Validation cascade based on F1 . . . 39
5.3 ROC curves for feature based models . . . 41
5.4 PR curves for feature based models . . . 42
5.5 Confusion matrix Cascade 40 . . . 44
5.6 Confusion matrix Multinomial Logistic Regression . . . 44
5.7 glad004 and glad017 postprocessing swimmer plots and spectrograms . . . 45
5.8 glad132 and glad105 postprocessing swimmer plots and spectrograms . . . 45
5.9 Raw EEG sample from glad064 . . . 46

A.1 Training loss: batch size 128, shuffle . . . 65
A.2 Training loss: batch size 256, denseblock 7 . . . 66
A.3 Training loss: batch size 256 . . . 66
A.4 Training loss: batch size 128, learning rate 0.1 . . . 67
A.5 Training loss: batch size 128 . . . 67
A.6 Training loss: batch size 256, no transfer learning . . . 68
A.7 Training loss: batch size 256, full model trained . . . 68

Introduction

Artificial Intelligence (AI) has been deemed a transformative technology with immense potential in the field of medicine. Through machine learning and deep learning, AI is capable of supporting medical specialists with disease detection [67], which is of immense value for healthcare and results in an increase in overall patient well-being [54]. Early and accurate disease detection provides significant societal benefits. Firstly, it significantly enhances patient outcomes across various medical fields [53]. This approach enables healthcare providers to identify and address potential health issues before they become more severe [5] [83]. Secondly, diagnostic support systems hold the potential for enormous economic benefits [89] [61] through reduced healthcare costs and decreased diagnostic mistakes.

Advancing disease diagnosis with AI is essential for several reasons. Medical specialists have limited time [71], diseases can evolve, and patient dynamics change [99], making clinical interpretation of medical information a cognitively challenging task and causing diagnostics to be prone to human errors [54]. The increasing pressure on medical specialists is likely to persist due to projected workforce shortages among healthcare professionals, driven by an aging population and a rise in chronic disease [21]. Research in AI models for medical diagnostics has already shown that training models on large-scale data can yield better diagnostic performance than experts and significantly improve the experts' performance when aided with AI diagnostic support systems [41] [49].

Many of the various medical specialties can greatly benefit from diagnostic AI, and this is particularly true for Neurology and Electroencephalography (EEG) analysis. There are three main reasons, that fall under the previously made arguments, why this specialization would benefit greatly. First, early detection of diseases have a great influence in patient outcomes. EEG analysis is a crucial tool for diagnosing epilepsy and seizure disorders [77] [58], monitoring brain activity in Intensive Care Units (ICUs) [3], and evaluating neurological conditions such as acute brain injury [10], strokes [91], and brain tumors [51]. It is also essential in diagnosing dementia [2] and sleep disorders [8]. The noninvasive nature of EEG and the ability to do real time monitoring make this method invaluable in clinical and research settings.

Second, there is a shortage of neurological expertise globally [50]. This is also projected to become worse as the widespread presence of neurological diseases increases. Given that people with epilepsy make up 1% of the world population and the fact that morbidity can be prevented with cost effective medicines [52] [7] there is an urgent need to use AI for improved patient outcomes in neurological care.

Third, EEG interpretation is difficult, and clinical decision making based on EEG is susceptible to bias which causes misinterpretation and misdiagnosis [34] [35] [60] [4]. This difficulty is caused by the non-stationary, non-linear, and non-Gaussian nature of EEG signals and brain signals [36] [78], which are grounds for the variability in the signal's statistical characteristics over time. Additionally, variations in electrode sensor placements, differences in EEG acquisition hardware from various manufacturers, and even minor discrepancies in electrode application on the skull can significantly affect EEG signal readings. These variabilities underscore the substantial challenges in achieving consistent and generalizable EEG signal interpretation across different individuals and sessions. Furthermore, manual visual inspection and analysis of EEG is time consuming and takes years of clinical training [73]. Inspection and monitoring of long-term medical EEG signals is especially prone to human errors because of the requirement for a high level of focus over this longer period of time.

The analysis of Ictal-Interictal Injury Continuum (IIIC) patterns is crucial in Neurology and ICU settings, due to their association with critically ill patients and increased mortality rates [33] [96] [Table 1]. IIIC patterns consist of seizures (SZ), lateralized and generalized periodic discharges (LPD, GPD), and lateralized and generalized rhythmic delta activity (LRDA, GRDA). These patterns can lead to significant brain damage and elevate the risk of in-hospital fatalities [43] [96] [11] [63]. Central to patient management is the identification of IIIC patterns, as they play a key role in guiding treatment decisions. Although aggressive medicine can suppress harmful brain activity, the primary goal remains to preserve brain function while minimizing adverse effects [75]. It is crucial for clinicians to distinguish which patterns are more closely related to seizures and which correlate more strongly with mortality risks [33]. Understanding and identifying these patterns not only aids in patient care but also drives research into

new treatments and gives insight into seizure mechanisms. Automated identification of IIIC patterns AI models would ensure targeted treatment leading to more effective interventions, improved patient outcomes, and reduced healthcare costs.

Table 1: EEG Patterns and Their Clinical Associations

Pattern	Indication	Associated with Seizure	% of ICU Patients	Mortality
LPDs	Acute stroke, traumatic brain injury, encephalitis, tumors	Yes	8.7%	24-41%
GPDs	Toxic–metabolic encephalopathy, anoxic brain injury, acute brain injury, infections, epilepsy	Yes	0.8-4.5%	30-64%
LRDAs	Intracerebral hemorrhage, subarachnoid hemorrhage	Yes	4.7-7.1%	Not specified
GRDAs	Variety of cerebral lesions and metabolic disturbances	No	Not specified	Not specified

1

Project and Research Description

1.1. Related work

There is an extensive body of work for machine learning in seizure detection. Past research on general seizure detection includes both traditional machine learning and deep learning methods. K-nearest neighbors were used to distinguish seizure from non-seizure EEG data [44], wavelet packet decomposition and random forest were combined to classify signals after feature importance analysis [79] and wavelet transform was commonly used for EEG feature extraction [76] [84] [23] with support vector machine being the preferred classifier [28] [46] [14]. Developing accurate seizure detection models with these traditional machine learning methods requires expertise in signal processing and data mining, showing high efficiency on small datasets but encountering limitations as data size increases [92].

Because of this problem, deep learning models have been employed to achieve better results in the field of seizure detection on EEG data. Haidar Khan et al. applied the Wavelet Transform (WT) for signal transformation and analyzed changes in probability distributions using Convolutional Neural Networks (CNN) and Kullback-Leibler divergence (KL divergence) for data probability distribution methods [38]. Kostas et al. trained a Long Short Term Memory (LSTM) model to capture temporal information for their predictions [82]. Liu et al. introduced a patient-independent approach in epilepsy research by deploying a Bidirectional Long Short-Term Memory (Bi-LSTM) network [47]. Recent studies have also explored CNN-based deep learning models, including 3D-CNN and ResNet, for classification purposes [62], [95], [29]. Lee et al. worked on a hybrid approach combining ResNet50 and LSTM with supervised contrastive learning [45].

These developments provide impressive results in seizure detection but they are all missing the ability to detect LPDs, GPDs, LRDAs and GRDAs. This is mainly because there are no publicly available datasets that include these patterns. To solve this problem, Jin Jing et. al started by doing research on rapid IIIC annotation in order to create large datasets for training machine learning models. They utilized unsupervised machine learning to demonstrate that EEG data can be efficiently clustered into distinct patterns, significantly enhancing the labeling process for IIIC patterns. This method proved to be 60 times faster than manual review [31]. Their approach began with the application of a Change Point Detection (CPD) algorithm to analyze the frequency characteristics of the signal, thereby segmenting the EEG. In parallel, time and spectral features were extracted. The dimensionality of these features was reduced per-patient and then clustered to form a dictionary. This dictionary was then combined with the segments to create histograms. These histograms were further clustered and subsequently scored by EEG experts. A follow up study showed that the pairwise agreement from this automated method was not significantly different from inter-rater agreements using manual labeling [32].

Subsequently, their attention shifted to gathering large annotated datasets in order to train deep learning models and achieving expert level classification on IIIC patterns. Methods focused on overcoming challenges like inter-rater agreement, substantial class imbalance in datasets, extracting features from raw EEG and leveraging unlabeled data. Wendong et al. used deep active learning on a large labelled datasetset from 1454 hospitalized patients. Their method consisted of first training a DenseNet Convolutional Neural Network (CNN) to create a 2D embedding map and using Nearest-neighbor label spreading to create additional pseudo-labels for a second round of training. Results showed label

spreading increased convergence speed and models approached expert level performance across IIIC pattern categories [19]. This research resulted in a follow up paper where another CNN based on the DenseNet architecture named Sparcnet was trained using 6,095 scalp EEGs from 2,711 patients, both with and without IIIC events [30]. The training and testing datasets comprised of 50,697 distinct or separate segments that were selected on specific criteria. These EEG segments were annotated by 20 experts. The research aimed to evaluate Sparcnet's performance in terms of sensitivity, specificity, precision, and calibration against that of experts for identifying IIIC events. The model was evaluated on the calibration index, the the model's receiver operating characteristic (ROC), and precision-recall curves (PRC) and achieved scores comparable to the experts' benchmarks for the six IIIC and the non IIIC class.

Despite these advances in IIIC detection, models are not yet capable of replacing Neurologists because there is a need for high quality data in order to build robust models [65]. In general seizure and IIIC detection, few published algorithms have fully addressed the requirements for successful clinical transition due to three major shortcomings [56].

1.2. Problem Definition and Research Questions

1.2.1. Introduction

This research aims to take steps in developing Seizure and IIIC detection models on long and noisy EEG recordings from critically ill ICU patients because automated detection can significantly help neurologists and patients. This is a relatively unknown field of research. The goals are to reduce the false positive rate, overcome data limitations, and improve the generalization to continuous EEG data which can be considered as significant challenges. The previously named challenges are addressed by building on the foundational work of Jin Jing et al. and applying transfer-learning with the Sparcnet model on extensive, continuous ICU EEG recordings from critically-ill patients that were acquired by Massachusetts General Hospital. The data consists of raw EEG recordings measured in microvolt with a 10-20 international system from 140 patients that are processed to 10-second L-Bipolar 16 channel segments with labels from three experts. The labels consist of the seizures, IIIC patterns, and the "Other" class which consists of both normal brain activity and artifacts, noise that is not relevant for diagnosis.

This study specifically investigates the effectiveness of Dense Convolutional Neural Networks in detecting Seizure and IIIC patterns within this continuous and noisy data. The variability in ground truth labels from experts is inherent to the nature of EEG signals. Consequently, the previously mentioned Bag of Words method, refined by an expert neurologist, is used to enhance data quality. Because of the success in reducing false positive rates on a smaller set of patients [15] and due to their less complicated implementation, this research uses a feature based Boosting Cascase and Multinomial Logistic Regression to post process the outputs from Sparcnet. These efforts set the stage for the following problems, knowledge gaps and research questions:

1.2.2. Problem 1 : Limited High Quality Data Availability

Data availability is limited, particularly for IIIC patterns, which hinders the development of robust models for this classification task. Most datasets for seizure detection have fewer than 25 subjects and feature different sampling frequencies, segmentation methods, number of channels, and placement methods [90]. There is also significant variability in ground truth labels. The average percent group consensus among experts is 65%, due to variations in decision thresholds [30]. If there is ambiguity and no clear consensus on the labels, machine learning models will struggle to learn the correct patterns. These shortcomings are especially prevalent in the emerging challenge of the Ictal-Interictal Continuum, as there are no publicly available datasets for these patterns.

This underscores the necessity of researching techniques for rapid annotation of EEG segments and developing methods capable of learning from smaller datasets. It highlights the need to address the limited data availability of IIIC patterns and to enhance the quality of labels. Such advancements are essential until large high quality annotated datasets exist and in order to realise models that are able handle real-world data and deliver classifications at an expert level under current circumstances.

A bag of words method has the potential to address this concern. The method revolves around finding unique words and using their counts in text documents to make vector representations of those documents. This vector representation can then be used to classify documents. The method which

stems from Natural Language Processing and Information Retrieval domain has crossed over two times to EEG data. Wang et al. showed that a bag of words representation can effectively capture both local and global structure similarity information in the EEG signal [87]. Furthermore, as stated in the related work section, Jing et al. showed that combining Bag of Words with Change Point Detection can be used to label EEG segments faster. How Bag of Words can be applied in the problem of Seizure and IIIC classification is fully described in section 3.4. This problem and method require further research and qualitative analysis on long EEG recordings from patients in the ICU, posing the first research question:

RQ1.
To what extent can Change Point Detection, K-means Clustering and Power Spectral Density be used as a Bag of Words method to improve EEG labels on continuous data from 140 patients?

1.2.3. Problem 2 : Reduced performance on noisy continuous EEG-data

Deep learning models that process EEG data have not been fully tested on their generalization ability because they are typically trained on the limited database of scalp EEG recordings [38]. Researchers conclude that methodologies need to be extensively tested in clinical practice on more EEG data than for example widely used datasets like CHB-MIT [82] [62]. Reduction in performance can be observed when changing test datasets, suggesting that deep learning models tend to overfit to the training data [94]. This lack of generalization is often caused by models that learn patient specific patterns which is noticed when evaluating on the test set that has new patients [20] [85]. This effect of reduced performance on a new test set is especially pronounced when applying models like Sparcnet to real-world continuous data for IIIC detection, which is noisier than most sampled data [15].

This is why there is a need to extensively validate these models on large scale real world continuous EEG datasets from a large number of critically ill patients. By applying them to this type of data we can make sure that they are robust in the clinical setting and can deal with different patients, a high variability of seizure patterns, non pathological EEG background rhythms and artifacts. Training and testing on this type of recording gives a sense of the capability from deep learning models to really assist in helping Neurologists and what is needed to develop these models further.

In order to improve generalization and deal with the limited data availability for IIIC patterns introduced in the previous section, Transfer Learning can be used. Transfer learning is a deep learning technique that uses a dataset with a similar distribution and task in order to initialize learning for training on another dataset [88]. A straightforward transfer learning approach that has proven to work with medical data is fine tuning [40], where parts of the network from a pretrained model are frozen and other parts are retrained. It has three main advantages that can be leveraged. First, it is able to enhance model generalization by using a pre-trained model such as Sparcnet trained on a large dataset and prevent overfitting. Second, the models performance on the smaller dataset can be improved by also benefiting from the features learned on the large dataset. Third, it can reduce training time and data need because the model has already learned relevant features for IIIC detection [72] [97]. Since Transfer Learning deals with both problem one and two it is subject to the second research question:

RQ2.
How well can Transfer Learning contribute, in Dense Convolutional Neural Networks, to capture better the complex nature of extensive ICU EEG data and their IIIC patterns?

1.2.4. Problem 3 : High False Positive Rate

The number of false positives on datasets of real longer continuous EEG recordings is generally too high. Research by Dirks et al. demonstrated that Sparcnet struggled with a high false positive rate on noisy continuous data. Sparcnet particularly classified samples as seizures that could be clearly identified as artifacts. A large number of false positives can lead to unnecessary interventions from neurologists. Rather than alleviating the time constraints faced by neurologists, these models could do the exact opposite.

This last problem emphasizes the requirement for IIIC detection models to have the necessary post processing steps that can reduce these misclassifications. The false positive rate of Sparcnets predictions can be reduced by two postprocessing methods that use manually engineered features. This research proposes to use a Boosting Cascade and Multinomial Logistic regression. The cascade was

trained by choosing features and thresholds that could keep a high level of true positives while reclassifying samples that were not IIIC patterns. This method leverages building a strong classifier from multiple weak classifiers and rapid rejection of negatives. Multinomial Logistic Regression is a statistical model that can predict multiple classes generalized from the binary classification method Logistic Regression and is further descriped in section 5.2. Both the Boosting Cascade and Multinomial Logistic regression models are used to recalibrate the output from Sparcnet. As opposed to computationally intensive and time-consuming deep learning experiments these methods take less time to train and can be implemented more easily.

RQ3.
To what degree can a threshold based Boosting Cascade and Multinomial Logistic Regression with feature engineering reduce the false positive rate of the IIIC pattern detection?

1.3. Outline

This work is divided into the following chapters:

Chapter 2: Background

The background chapter serves to further expand on concepts used in chapters 3, 4, and 5. The chapter discusses the EEG signal and its spectral content in section 2.1 in order to give context for the spectrograms that are used throughout the chapters. It continues to expand on IIIC patterns and their characteristics in section 2.2, a background on Convolutional Neural Networks as a preliminary for the DenseNet model in section 2.3, and finally explains the idea of the DenseNet model, that is used in the transfer learning experiments, in section 2.4.

Chapter 3: Bag of Words and Change Point Detection

This chapter describes the methods that were used to improve the label quality of the EEG segments from the 140 patients dataset. The chapter starts off by showing the problem of label quality which the method tries to solve in section 3.1. After this the theory of Change Point Detection, K-means clustering, and how they are used in the Bag of Words method are explained in sections 3.2, 3.3, and 3.4. The results of the the full method are presented in section 3.5, the implications of these results are discussed in section 3.6, and finally the limitations of using this method to create new labels are discussed in section 3.7.

Chapter 4: Transfer Learning with Dense Convolutional Neural Networks

This chapter presents multiple transfer learning experiments that were done with the new labels from the previous chapter. The goal of using these methods is to outperform the Sparcnet 1 model. It opens up with a description of the data, how the data was processed, the loss function used to deal with data imbalance, and the computational setup in sections 4.1, 4.2, 4.3, and 4.4. The chapter then continues by describing the transfer learning experiments and the metrics used to evaluate them in sections 4.5 and 4.6. Finally, it ends with a description of the results, a discussion of the results, and the limitations of using transfer learning to learn complex patterns from continuous EEG data in sections 4.7, 4.8, and 4.9.

Chapter 5: Feature Based Post Processing

Deep learning experiments require a lot of tuning and time to train so in order to provide an alternative this chapter presents postprocessing with feature based Boosting Cascade and recalibration with Multinomial Logistic Regression. Both methods use features that say something about the frequential, morphological, and temporal information of the EEG signal. The features are used to recalibrate the output from the Sparcnet 1 model. The chapter begins by explaining the theory behind the Boosting Cascade and Multinomial Logistic Regression in sections 5.1 and 5.2. It then concludes with a description of the results, a discussion on what to conclude from these results, and the limitations of both methods in sections 5.3, 5.4 and 5.5.

Chapter 6: General Discussion

The general discussion aims to summarize the results and limitations from chapters 3, 4, and 5, put the research in the context of the literature and discuss steps that can be made in future work. The chapter opens with a summary of the results and limitations in sections 6.1 and 6.2. It then continues with the context to related work in section 6.3. Finally, the chapter concludes with the recommendations for future work in section 6.4.

2
Background

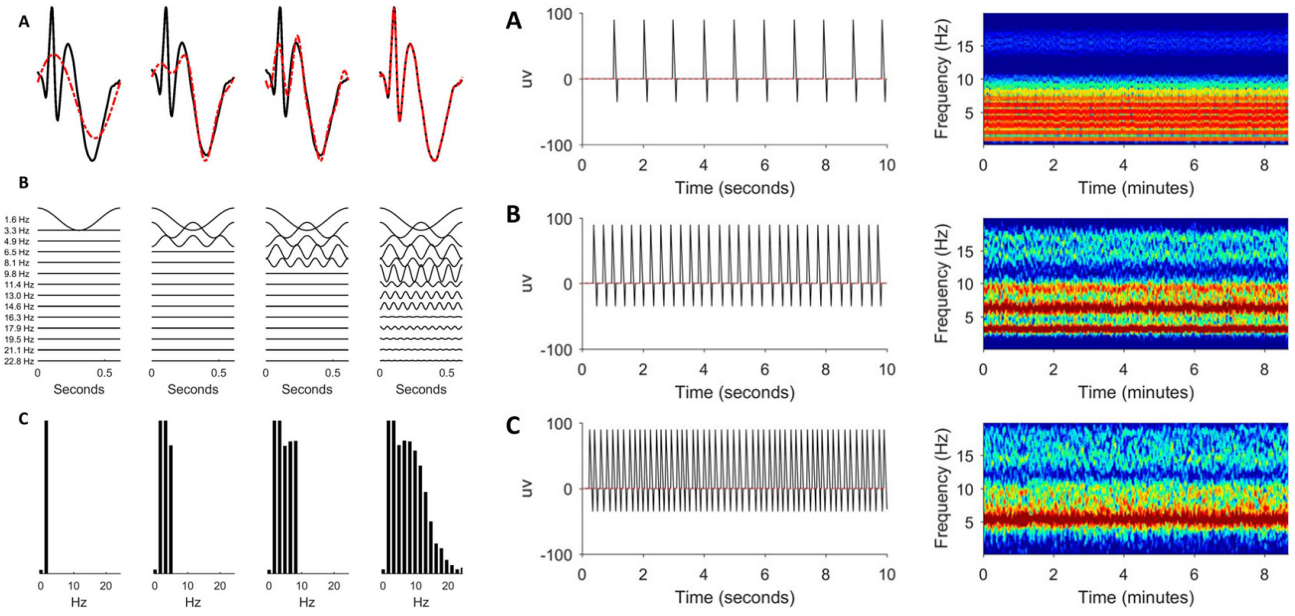
This chapter serves as background knowledge about EEG signals, IIIC patterns, Convolutional Neural Networks, and the DenseNet architecture. The frequency content of the EEG signal is explained and will be used to create features for the postprocessing models in chapter 5 and qualitative analysis of the outputs of the models that are researched. The IIIC patterns and their specific characteristics that are at the center of the classification task are further detailed. Finally, the mechanics of Convolutional Neural Networks and the DensNet architecture are explained which form the basis of the Transfer Learning experiments in chapter 4.

2.1. Electroencephalography

Inside our brain, neurons have an electrical charge. Due to brain activity, this charge changes. As the cortex is folded, the direction of these charges typically align, and a net potential can be measured outside the skull. This signal is referred to as EEG signal and is typically sampled by non-invasive electrodes placed on the scalp often with an International 10-20 system. Measuring changes in electrical activity results in EEG waveforms. The strength of the fluctuations is measured in microvolt and the signal can be decomposed into a complex mixture of waveforms with different frequencies. The frequency of a waveform is defined by how often the wave repeats itself over time. An EEG signal can then be decomposed from the time domain to the frequency domain with a Fourier transform (equation 2.1) to determine the power of the different frequencies in the signal. In the Fourier Transform, the amplitude over time $x(t)$ is integrated over all time t while being multiplied by a complex exponential containing the frequency (f), time (t), and an imaginary unit (i). The small expression that is $x(t)e^{-2\pi ift}$ encapsulates the idea of winding a graph around a circle with a variable frequency. This graph that is wound up around a circle for this specific frequency contains a center of mass that is larger for frequencies that are present in the original circle. This process actually creates a new function that has frequency as an input and converts the time-domain signal $x(t)$ into the frequency-domain representation $X(f)$, revealing the signal's frequency content. This results in the power or amplitude of each frequency component (figure 2.1 B & C).

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-2\pi ift}\,dt \tag{2.1}$$

The frequency content of a signal is important for characterizing normal and abnormal brain activity [27] [70]. The frequency bands that are most important for clinical decision making are alpha (8-12 Hz), beta (12-30 Hz), theta (4-7 Hz), gamma (30-100 Hz), and delta (0.5-4 Hz) waves [6]. Alpha waves can be associated with seizure control in epilepsy [1], slow waves like delta waves are of prime importance of detecting epilepsy [86], and ratios between for example the alpha and delta waves can be used as features in detecting harmful brain activity [98]. A central tool in quantitative EEG that is standard in most quantitative EEG software is the "spectrogram" [57]. The spectrogram plot has time on its x-axis, frequency on the y-axis, and power in color (figure 2.1b A, B & C) and is essentially a plot of the Fourier transform over time. These spectrograms are widely used by neurologists and relate back to information in the raw EEG signal. Through spectrograms, clinicians can infer underlying EEG patterns.



(a) Waves to frequency bins **(b)** Waves to Spectrograms

Figure 2.1: Waves and their frequency content [57]

Seizures for example often have a gradual increase in the power of higher frequencies which is more difficult to see in the raw EEG signal but can clearly be observed in the spectrogram (figure 2.2 C & D).

A better way to make spectrograms is by using Multi-taper spectral estimation [37]. The power spectral density (PSD) that is obtained by the signal's raw Fourier transform contains variance, and noise and is a biased estimate of the true spectral content. This is caused by the spectral leakage that occurs when using a taper on finite data. The taper is also called a window function and has the property of being zero valued outside of an interval. Multi-taper spectral estimation makes use of multiple window functions and averages their single taper spectra across those tapers to create less noisy spectrograms. The tapers that are being used come from the discrete prolate spheroidal sequences (DPSS) and construct low-bias, statistically consistent spectral estimators that reduce spectral leakage.

2.2. IIIC patterns

The ictal-interictal continuum (IIC) consists of a spectrum of EEG patterns that are positioned between clearly ictal (seizure) and interictal (between seizures) patterns. These patterns share some features with seizures but can not be definitively classified as seizures. As mentioned in the introduction there are two types of patterns next to seizures and those are Periodic Discharges (PDs) and Rhythmic Delta Activity (RDA). These two forms of patterns can appear across the entire brain resulting in generalized periodic discharges and rhythmic delta activity (GPD, GRDA). In contrast when these patterns can appear in one of the two hemispheres which are called lateralized periodic discharges and rhythmic delta activity (LPD, LRDA). The patterns can be seen in figure 2.3.

PDs emerge as waveforms that repeat with a consistent shape and lasting time, each set apart by noticeable intervals which can be called the inter-discharge interval separating the discharges as can be seen in Figure 2.4. These waveforms show up at almost consistent times, where the definition of "nearly consistent" is the cycle length altering by less than 50% in most pairs of cycles. A discharge is marked by waveforms that do not go beyond 0.5 seconds in duration, regardless of the phases involved, or those extending 0.5 seconds or more but limited to three phases or fewer [24]. RDA is characterized by the repetitive occurrence of waveforms that share a uniform appearance and duration but unfold without any inter-discharge intervals between subsequent waveforms. The time frame, or period, of these rhythmic occurrences, should remain under a 50% variability from one cycle to the following in the majority of cycle pairs. Although a sinusoidal waveform is often seen, the scope for what is considered rhythmic includes patterns that may exhibit sharpness at their peaks or troughs. For a pattern to fall under RDA, it needs to be within a frequency range of 0.5 to 4.0 Hz. Rhythmic and

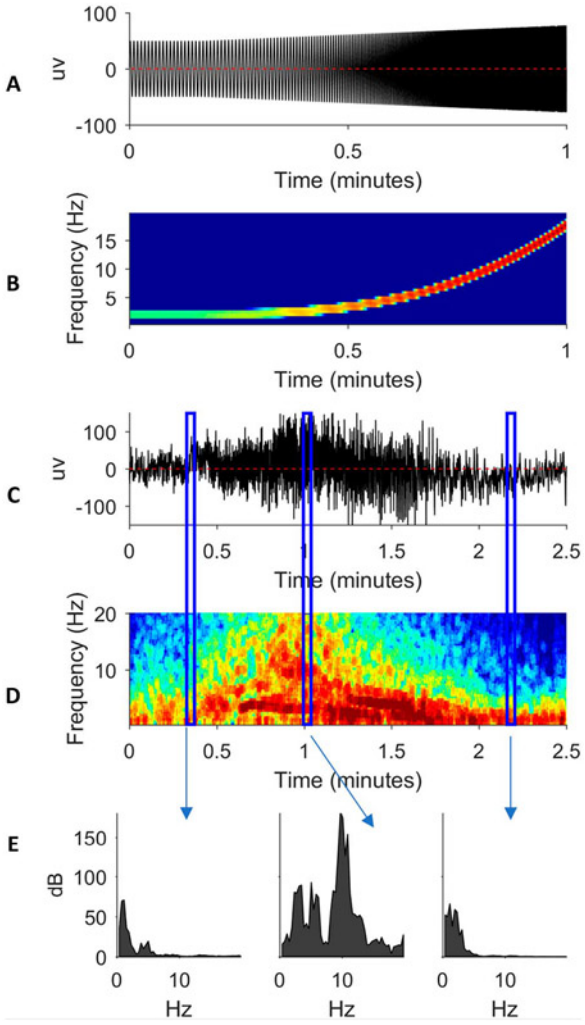


Figure 2.2: Spectrogram of real seizure [57]

periodic patterns need at least six cycles or discharges in a row in order to qualify.

LPDs are the most commonly observed pattern and are seen in 8.7% of critically ill patients [81] [12] [9]. Often they are seen in patients with structural brain injuries such as acute stroke, traumatic brain injury, encephalitis, and tumors [9] [17]. They are associated with an increased risk of seizures and can appear in the setting of epilepsy, systemic infection, metabolic and toxic insults as well as in the absence of structural lesions [17]. GPDs were observed in 0.8-4.5% of critically ill patients [81]. The most common etymologies are toxic–metabolic encephalopathy, anoxic brain injury, acute brain injury, infections, and epilepsy. They are also often associated with seizures particularly nonconvulsive seizures and nonconvulsive status epilepticus [16] [64]. LRDAs are perceived in 4.7-7.1% of critically ill patients that are monitored by continuous EEG [81] [18] [69] [74]. Intracerebral hemorrhage and subarachnoid hemorrhage are the most common causes of LRDA [18]. GRDAs are seen together with a variety of cerebral lesions and metabolic disturbances [81] [74]. Unlike the previous patterns, GRDA does not seem related to seizures [42] [80].

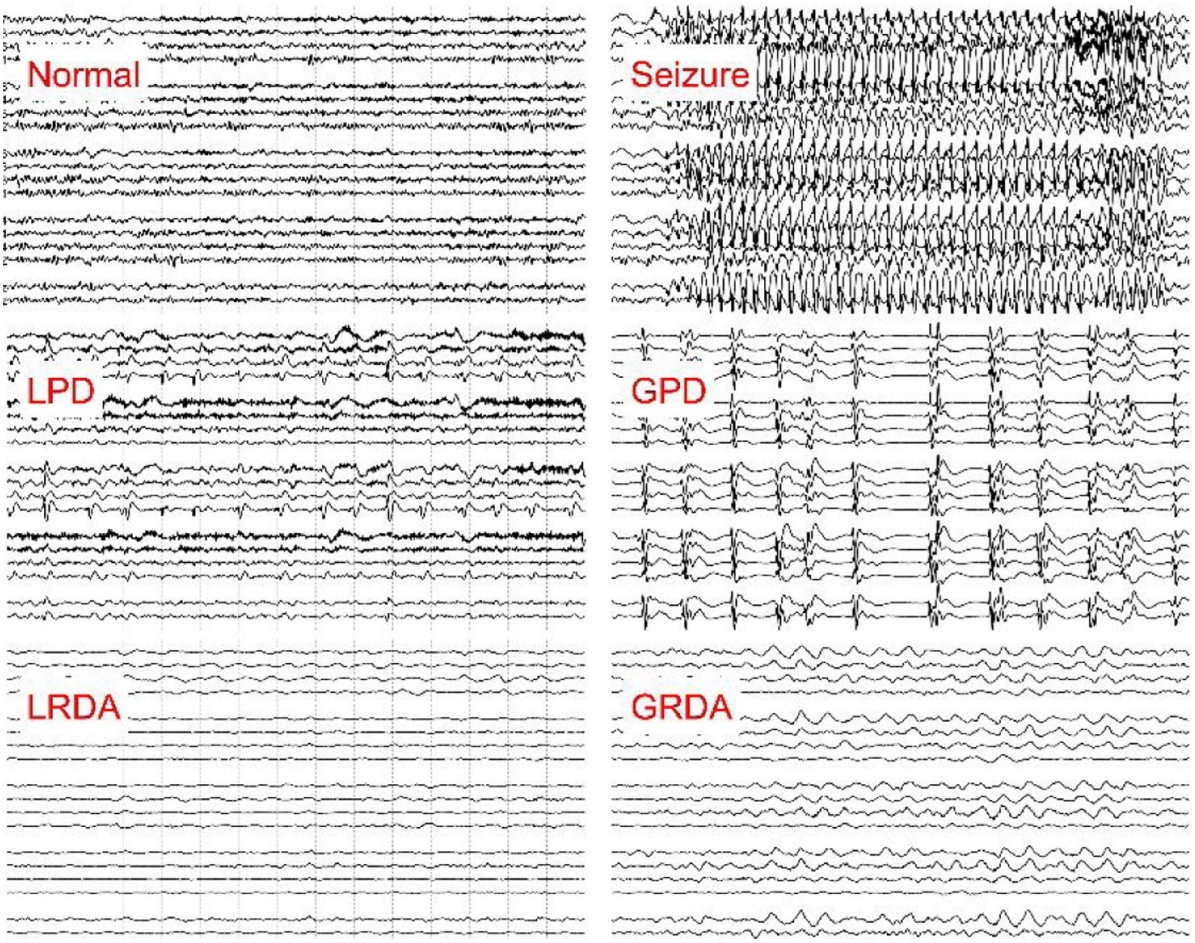


Figure 2.3: IIIC patterns

2.3. Convolutional Neural Networks

In this thesis, Convolutional Neural Networks (CNN) like the Sparcnet model are used to do transfer learning and apply post processing techniques. CNNs are neural networks specifically designed for processing data with a grid like topology [22]. These networks originated from the field of computer vision and were designed to be able to get an understanding of 2D or 3D image data. As opposed to classical neural networks they preserve spatial information by not just flattening the input of the image. They are capable of extracting high level features like edges or other shape patterns. By using a kernel, CNNs reduce the size of the image to a form that can be easily processed by the network. The kernel consists of weights that can be learned through a loss function and the parameters are optimized via gradient descent-based optimization algorithms. Unlike general neural networks, which use matrix multiplication, CNNs have a special convolutional operation present in one of the layers of the network. The convolution operation which involves the weighted sum of products between the kernel and overlapping patches of the input produces a feature map. This feature map shows the intensity of predicted features across the different regions of the input.(2.5). Because the same kernel parameters interact with all spatial locations of the input data this operation is computationally efficient and it also makes use of the spatial information.

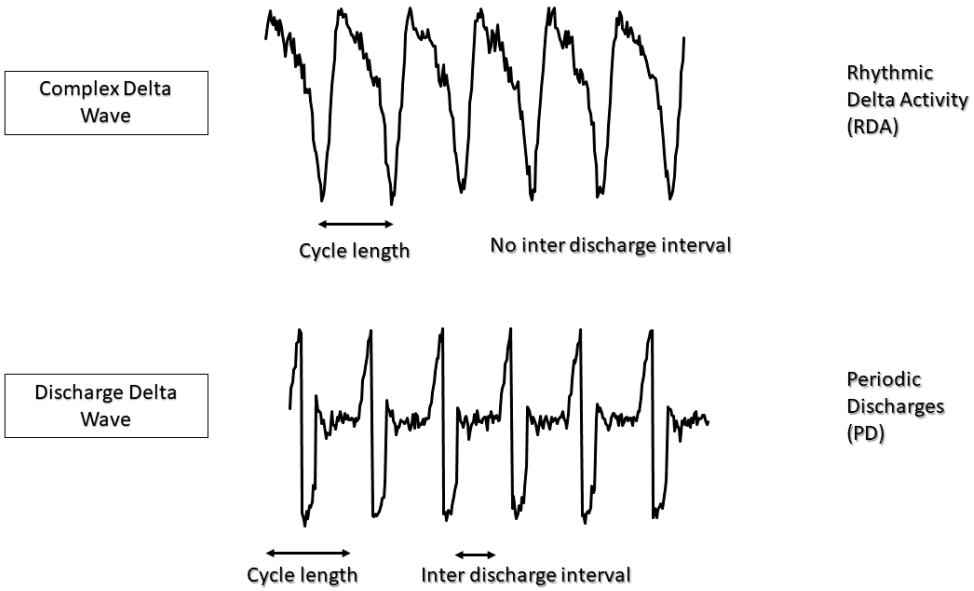


Figure 2.4: Rhythmic Delta Activity vs Periodic Discharges [24]

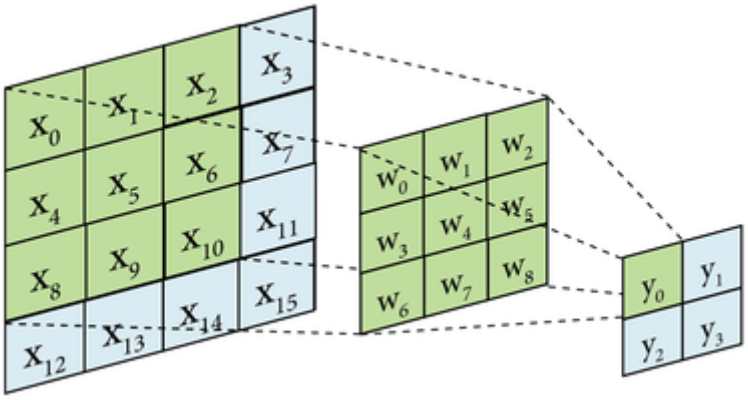


Figure 2.5: Convolutional operation [55]

In order to calculate the values y element wise multiplication and sum are applied as seen in equation (2.2). In the equation, I represents the input which in this case is a 2D matrix with indexes m and n. K represents the kernel with indexes i and j. The kernel K is usually smaller than the input matrix I and depending on padding and stride slides over a few data points from I.

$$S(i,j) \;=\; (I * K)(i,j) \;=\; \sum_{m}\sum_{n} I(m,n)K(i-m, j-n) \tag{2.2}$$

Applying many convolutional operations like this decreases the shape of the input (2.3). Continuously applying these operations will not leave any pixels left at the edges. One of the ways to capture the information at the edges is by using padding and adding zeros around the input(2.4).

$$\begin{aligned} \text{Shape} \;&=\; (n_h - k_h + 1) \times (n_w - k_w + 1) \\ &\text{with } n \text{ as input size and } k \text{ as kernel size} \end{aligned} \tag{2.3}$$

$$\begin{aligned}\text{Shape with padding } &= (n_h - k_h + p_h + 1) \times (n_w - k_w + p_w + 1) \\ &\text{with padding } p_h \text{ rows and padding } p_h \text{ columns}\end{aligned} \tag{2.4}$$

Another important parameter in the convolutional operation is stride which represents the number of rows and columns the patch is moved per slide. This can be done for computational efficiency or dimensionality reduction. The output shape becomes smaller based on the stride height s_h and stride width s_w (2.5).

$$\text{Shape with stride } = ((n_h - k_h + p_h + s_h)/s_h) \times ((n_w - k_w + p_w + s_w)/s_w) \tag{2.5}$$

After doing a convolutional operation the input is further transformed by adding a bias value and passing the input through a non-linear function just as with normal neural networks. Non linear functions like equations (2.6),(2.7),(2.8) and (2.9) [13] are also called activation functions. These functions introduce nonlinearity into the network, control the output range, and control the gradient flow when doing backpropagation during training. They are essential for neural networks by enabling them to learn complex patterns from the data. When doing the convolutional operation on an input that has for example three dimensions only the depth of the kernel changes but the operation is still in two dimensions.

Besides these operations and layers most CNN's have pooling layers to reduce data dimensionality and introduce spacial invariance. The goal of pooling layers is to reduce the number of trainable parameters while keeping the same relevant features even when the input is translated to a smaller extent [26]. Pooling techniques can for example consist of max-poling and average-pooling. As their name insinuates they can either take a max value out of an input patch or generate the average over that input patch.

$$\text{Sigmoid: } S(x) = \frac{1}{1 + e^{-x}} \tag{2.6}$$

$$\text{Tahn: } f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \tag{2.7}$$

$$\text{Relu: } f(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases} \tag{2.8}$$

$$\text{Softmax: } f_i(\vec{x}) = \frac{e^{x_i}}{\sum_{j=1}^{J} e^{x_j}} \quad \text{for } i = 1, \ldots, J \tag{2.9}$$

In a simple CNN, these layers are alternated and consist of the feature learning part of the network. Features learning layers are often followed up by a classification layer which flattens the features so that they can serve as input for a fully connected layer. This fully connected layer can end with a Softmax function that maps the output of the fully connected layer to a probability for a certain number of classes.

2.4. DenseNet

DenseNet is a form of CNN that, instead of making architectures more deep and wide, tries to leverage feature reuse to create a condensed model that is easy to train and highly parameter efficient. The main idea is to concatenate different feature maps that are learned by different layers [25]. In the DenseNet architecture, the input first passes through a batch normalization, convolution, and activation layer that generates the feature maps (equation 2.10). The input for the next layer is not only this feature map but the feature map and the original input concatenated. The input of each layer consists of the feature maps of the previous layers as in equation 2.11.

$$H(x) = \text{Conv}(\text{ReLU}(\text{BN}(x))) \tag{2.10}$$

$$x_l = H([x_0, x_1, \ldots, x_{l-1}]) \tag{2.11}$$

Having access to the raw information of the input and all the feature maps has a positive effect on the variation of the input between layers, mitigates the loss of information as the network deepens,

and prevents models from overfitting on noise from the training data. Because of the concatenation, the dimensions start exploding that is why after a few concatenations a transitional layer is added which consists of a 1D convolution layer in order to project the dimensions back to a lower dimensional one. A sequence of layers that concatenate feature maps before going to a transition layer is called a denseblock (2.7)

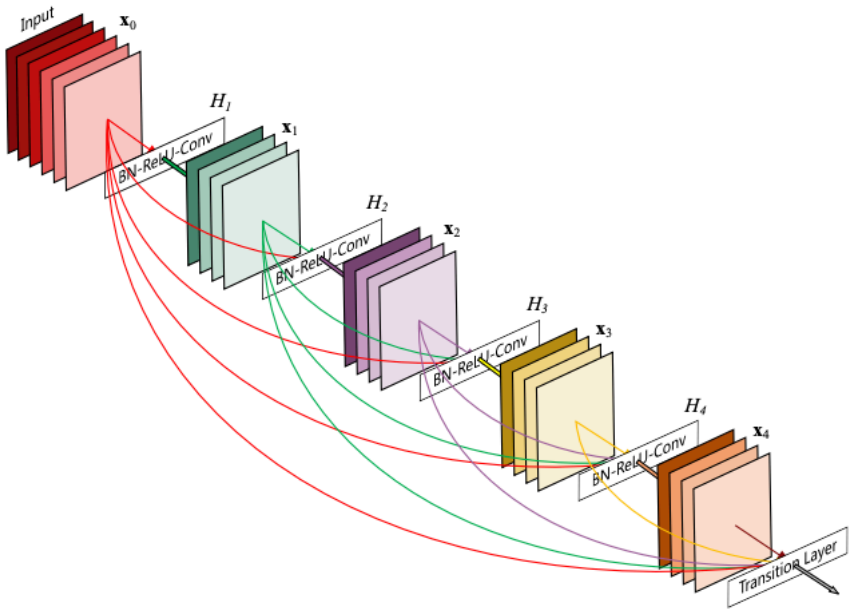


Figure 2.6: Denseblock [25]

An interesting effect that might be counterintuitive is that this approach actually requires fewer parameters. This is because through this method there is no need to re-learn redundant feature maps as the architecture promotes feature reuse. The DenseNet architecture differentiates between information that is added to the network and information that is preserved. DenseNet layers are really narrow and thus add a small set of feature maps to the collective knowledge of the network. The number of feature maps is also kept in check by a hyperparameter k called the growth rate (k_0 is the input dimension).

$$\text{Input feature maps of the l-th layer} = k_0 + k(l-1) \tag{2.12}$$

Another advantage is that DensNet architectures are easier to train compared to normal neural networks because each layer has direct access to the gradients from the loss function and original input signal because of the direct connections between layers. Finally, because of the short paths in the network, there is a strong regularization effect and the models tend not to overfit on smaller training sets.

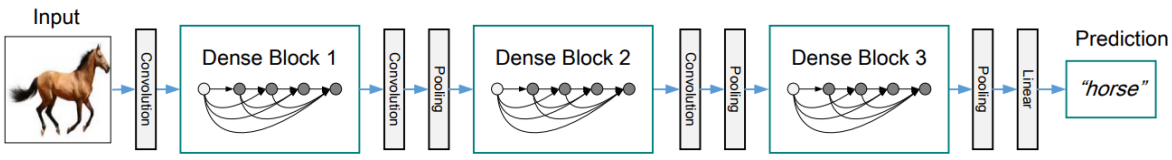


Figure 2.7: DenseNet architecture [25]

3

Bag of Words and Change Point Detection

This chapter deals with label quality and the methods described serve as a preprocessing step before applying the models in chapters 4 and 5. Bag of Words and Change Point Detection were used to improve the labels. One of the experts used a graphical user interface (GUI) to tune the parameters of the number of clusters in the k-means clustering of this Bag of Words in order to output better labels than the majority vote of the expert labels. In some of the cases, labels were better from the outputs of the models and in some cases, the labels were better from specific experts or combinations of experts and model outputs. Each case was reviewed by one of the experts and choices were made on which labels had the highest quality. The chapter starts by introducing the problem and by giving examples of the labels. Following this, the methods used to solve the problem are explained. Finally, some of the results and the limitations of these methods are discussed.

3.1. Label Quality

The opinions varied substantially between experts and not all experts produced labels for each patient. This problem of label quality can be seen in the examples from the dataset of a few sampled patients. Figure 3.2 shows segments of patients ”glad001” and ”goodSZ001” in the form of the expert labels, average power with change points and spectrograms. Both these patients show how different the labels can be for each expert. The classes are represented as swimmer plots with color codes seen in figure 3.1. The ”Other” class is represented as dark blue, the ”Seizure” class as red, the ”GPD” class as yellow, the ”LPD” class as orange, the ”LRDA” class as green, and the ”GRDA” class as light blue. The spectral content of the signal can be seen in the spectrograms denoted by the left lateral (LL), right lateral (RL), left para-central (LP), and right para-central (RP) brain regions and plots in the figures.

In glad001 there are no labels from expert A as seen from the grey bar in the A swimmer plots, expert B has the class ”LPD” as shown by the orange B swimmer plot and expert S has the classes ”Other”, ”LPD” and ”LRDA” which can be seen in the S swimmer plot based on the colors orange, green and dark blue. The majority of the labels end up being ”LPD” as seen by the orange color in the M swimmer plot where expert B’s labels overrule the tie in labels on some of the segments. The problem with this sample is that there are clear artifacts present in this sample which can be identified by the extremely high average power in the spectrograms for regions right lateral (RL) and right para-central (RP).

In the first half of the segment from goodSZ001 expert A sees ”Seizure” (red color bar in the A swimmer plot) while expert B detects ”LPD” (orange color bar in the B swimmer plot) and expert S detects ”Other” and ”LPD” (orange and blue bars in the S swimmer plot). In the second half of this segment, both experts B and S detect ”Seizure” (red bars in the B and S swimmer plots) while expert A thinks the ”LPD” class is present (orange bar in the A swimmer plot).

Figure 3.3 also shows the labels, spectrograms, average power, and change points but for patient ”glad022” and a different segment of patient ”goodSZ001”. This other patient and different segment from ”goodSZ001” give insight into the errors that can occur in the final labels when applying a majority

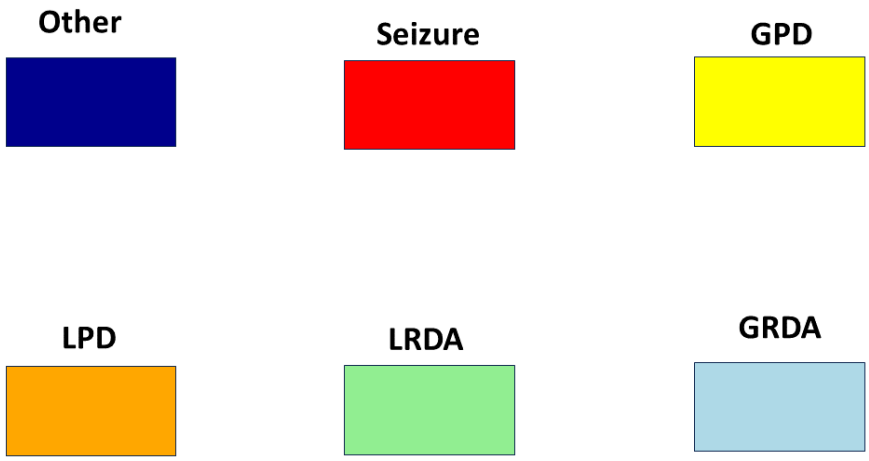


Figure 3.1: Color code of classes for swimmer plots

vote over the expert labels. In glad022 the expert labels from experts A and S are ”Seizure” (the red bars in the A and S swimmer plots) and the labels from expert B are ”GPD” (the yellow bars in the B swimmer plot) on segments where there is no signal. It is clear from the spectrograms in the left lateral, right lateral, left para-central, and right para-central that these sections should in this case be labeled as ”Other” because there is no measurement in the parts where the spectrogram is dark blue (0 dB). A measurement of 0 dB should be labeled as an artifact because there are no seizure or IIIC patterns in the EEG.

In the other segment from goodSZ001 experts B and S labeled the first part of the EEG as ”Seizure” (shown by the red color bars in the swimmer plots B and S) and expert A as ”LPD” (indicated by the orange swimmer plot A). This first part of the segment also shows really high average power just like in ”glad022” which also indicates an artifact.

3.2. Change Point Detection

Solving the previously mentioned problems is done through a pipeline that generates a number of new labels based on the expert labels and the power spectral density of the signal. The new labels that are generated are the clean labels denoted by C and the SZ labels which are a combination of the clean labels and the seizures from the experts. The pipeline starts by loading in the spectrograms of a patient and the labels from the experts. The spectrograms are calculated by using multitaper spectral estimation as described in the background section 2.1. The labels from the experts are used to create majority vote labels that are described in the swimmer plots as M. Since not every sample has a label from every expert there is a possibility for a tie. In this case, the label is overruled by the label from expert B.

The spectrograms are aligned with the labels and the mean power in decibels is calculated for each time point. Since this mean power can have a very high resolution that is very noisy it is smoothed using a Savitzky-Golay filter. The Savitzky-Golay filter is a generalized moving average with filter coefficients determined by an unweighted linear least-squares regression and a polynomial model. The degree of the polynomial is the smoothing parameter for the function and is set at 10 for all patients.

On the smoothed average power Change Point Detection (CPD) is applied. A changepoint is a moment in time when a statistical property of a signal abruptly changes. CPD is often used to find changes in time series signals [48]. The CPD algorithm used follows the following steps:

- Choose a point and divide the signal into two sections

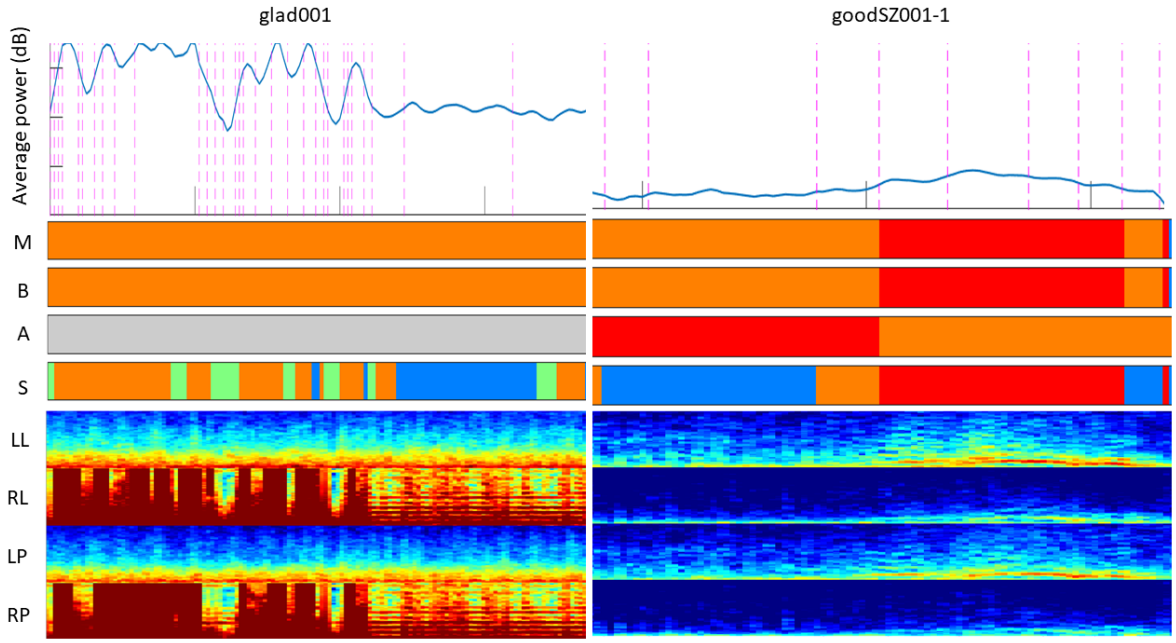


Figure 3.2: glad001 and goodSZ001: Expert labels, Average power (dB) and spectrograms for brain regions (LL, RL, LP, RP)

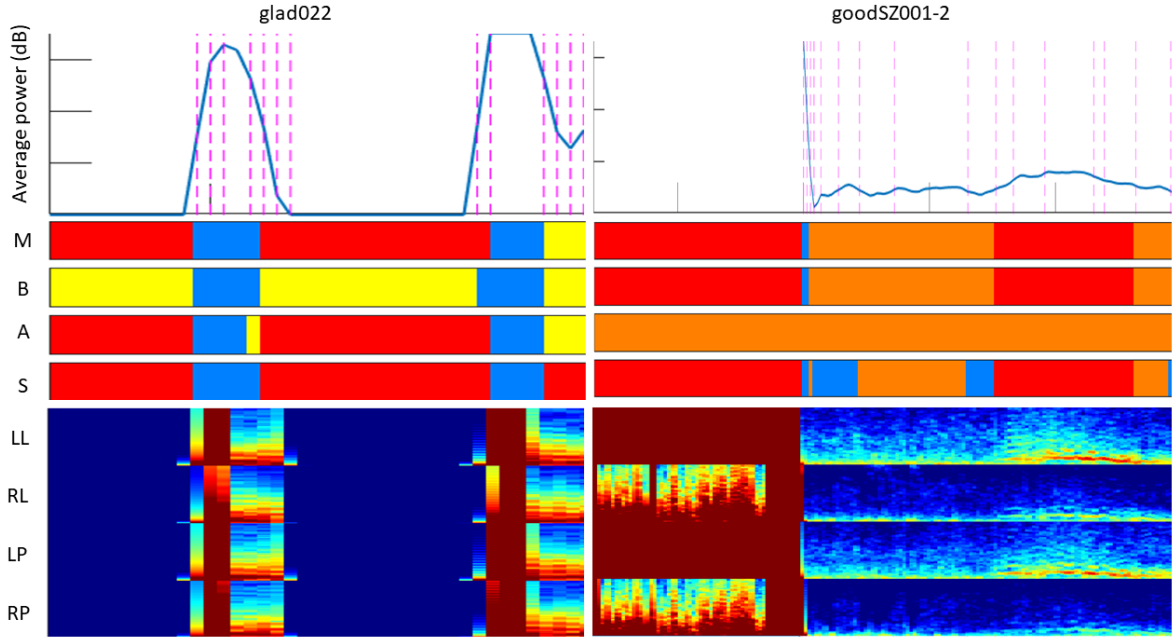


Figure 3.3: glad022 and goodSZ001: Expert labels, Average power (dB) and spectrograms for brain regions (LL, RL, LP, RP)

- Compute an empirical estimate of the mean power
- For each point add deviation from the empirical estimate
- Add deviations from section to section and compute the total residual error
- vary the location of the division point between sections until this residual error is the lowest

The empirical estimate, in this case, is just the mean of the mean power from segment k_r to k_{r+1} and given by equation 3.1

$$\langle x \rangle_{k_r}^{k_{r+1}} = \frac{1}{k_{r+1} - k_r + 1} \sum_{i=k_r}^{k_{r+1}} x_i \tag{3.1}$$

Theoretically now if every point becomes a changepoint the residual error vanishes which is why a penalty term is added that grows linearly with the number of changepoints. This means that the function rejects adding additional changepoints if the decrease in residual error does not meet the threshold. This means that given a signal $x_1, x_2, ..., x_N$ and changepoints $k_1, x_2, ..., k_K$ the error for the number of changepoints $J(K)$ is given by equation 3.2.

$$J(K) = \sum_{r=0}^{K} \sum_{i=k_r}^{k_{r+1}} \left(x_i - \langle x \rangle_{k_r}^{k_{r+1}} \right)^2 + \beta K \tag{3.2}$$

- $\left(x_i - \langle x \rangle_{k_r+1}^{k_{r+1}} \right)^2$ is the deviation of point x_i from the empirical estimate.
- βK is the penalty term, with β being a constant.

In summary, the average power signal is segmented in such a way that the deviation from the mean of those segments is as small as possible while preventing overfitting through a minimum threshold parameter. The optimization of the deviation error is done through dynamic programming and with early abandonment [39].

3.3. K-means clustering

The k-means clustering algorithm is used to make the Bag of Words representation of the data in order to spread labels to segments that have the same frequency content. K-means clustering is an unsupervised algorithm that is used to identify groups of data points and assigning each point to one of those groups with their centers. The algorithm tries to minimize the sum of the Euclidean distance between assigned cluster centers μ_i and the points x in the cluster C_i and the clusters k (equation 3.3).

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{3.3}$$

The algorithm is initialized by randomly assigning k initial cluster centroids. First, the nearest data points are assigned to the new cluster. In the assignment step, each data point x_p is assigned to the nearest centroid μ_i (equation 3.4). After reassigning the new labels the centroids are calculated again as in equation 3.5. These two steps are repeated until the centroids don't change any more.

$$C_i = \left\{ x_p : \|x_p - \mu_i\|^2 \leq \|x_p - \mu_j\|^2 \text{ for all } j, 1 \leq j \leq k \right\} \tag{3.4}$$

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x \tag{3.5}$$

3.4. Bag of Words

In the EEG recording K-means clustering is used on the power spectral density values to create a dictionary of "words". The entire signal and its frequency content are divided into segments with cluster centroids and labels. Then for each segment from the Change Point Detection, a histogram of cluster labels is made. Each time point in the sample that has dB values over all frequency bins is assigned to one of the clusters by comparing them to the cluster centroids. A histogram per sample is created from the counts of the cluster labels. These counts of cluster labels are the Bag of Words representation of the frequency content. The histograms are clustered again in order to spread labels to all cluster members. This process has two parameters the "number of words" for the first clustering and the "number of clusters" for the second clustering.

After these clustering steps cluster level labels are assigned to the entire cluster of histograms based on the majority label of the medoid. This way each cluster has a consistent label. As a final

step in the label spreading process, the labels are spread over the changepoint segments so that each segment that belongs to the same cluster has the same labels.

At this point, a postprocessing step was implemented to improve the labels. All samples that had average power outside of the range of [-20, 20] were set to the Other class. This resulted in cleaned labels C. Because the new cleaned labels missed some seizures that experts found in the data a third set of labels was created: any seizures in M were set in a copy of C creating SZ. This resulted in the labels M, C and SZ.

3.5. Results

Applying the previously mentioned pipeline on the examples from the label quality section resulted in the new SZ and C labels seen in figures 3.4, 3.5 and 3.6. In the figures the BOW color bar shows the clusters from the Bag of Words method and its corresponding sections.

The segment from glad001 in figure 3.4 shows that there are more ”Other” classes as seen by the dark blue bars in the regions where the average power spikes and the power over the frequency bins are high in the spectrograms. This is a clear improvement over the majority vote plot in figure 3.2. The segment of goodSZ001 in the same figure gave a similar result as opposed to taking the majority label (M) with the exception of the part after the seizure where it took over the label ”Other”. The segment from glad022 in figure 3.5 also shows an improved presence of the ”Other” class when the signal is zero and when there are abnormal power spikes as opposed to the ”Seizure” labels, depicted by the red color bars, that were seen in the M swimmer plot of figure 3.3. The SZ and C labels for the second goodSZ001 segment also considerably changed compared to M. The power artifacts at the beginning of the segment were relabeled as the ”Other” (the dark blue bars in SZ and C) class and showed correct ”Seizure” labels (the red color bars in SZ and C) towards the end where there was an increase of power in the left lateral and left para-central regions over certain frequency bins.

Figure 3.6 shows that in patient ”glad003” the parameters have a big influence on the C labels. The labels where the number of clusters was set to 30 classified the ”Seizure” classes from expert B as the ”GRDA” class, as shown by the light blue color bars in the C(50,30) swimmer plot, while a parameter of 50 resulted in the ”Other” class and label (the dark blue color bars). In this segment, there were only labels from expert B. This shows both the influence of the parameters on the C labels and that there are segments where the new C labels are not better than those of the experts.

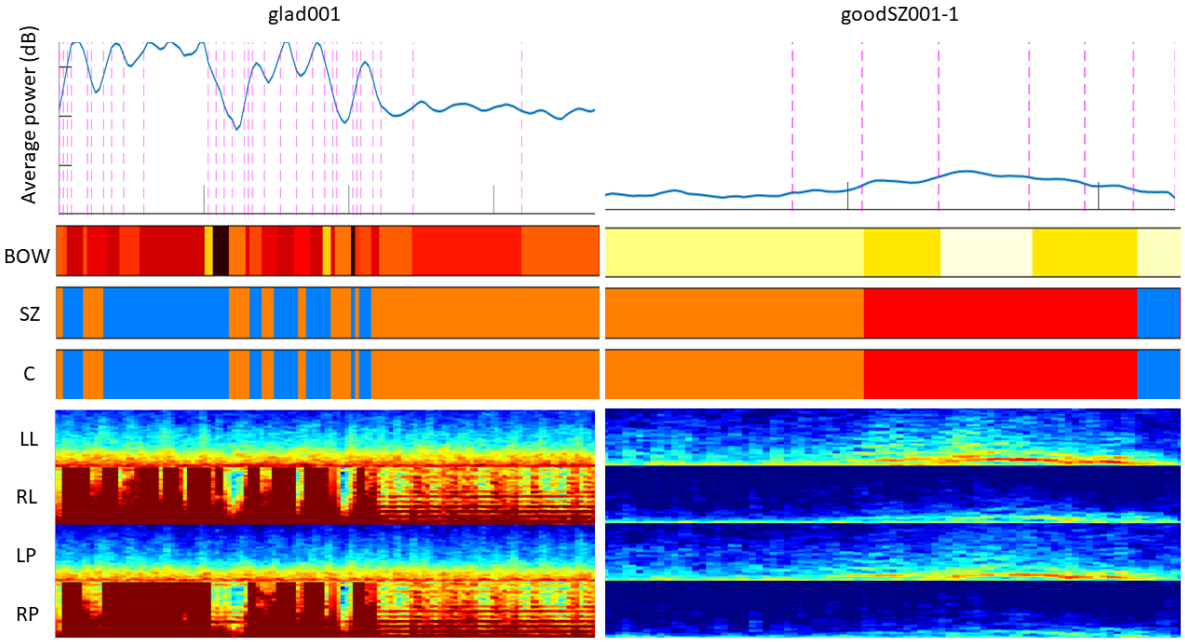


Figure 3.4: glad001 and goodSZ001: Bag of Words clusters (BOW), new labels (SZ, C), Average power (dB), for brain regions (LL, RL, LP, RP)

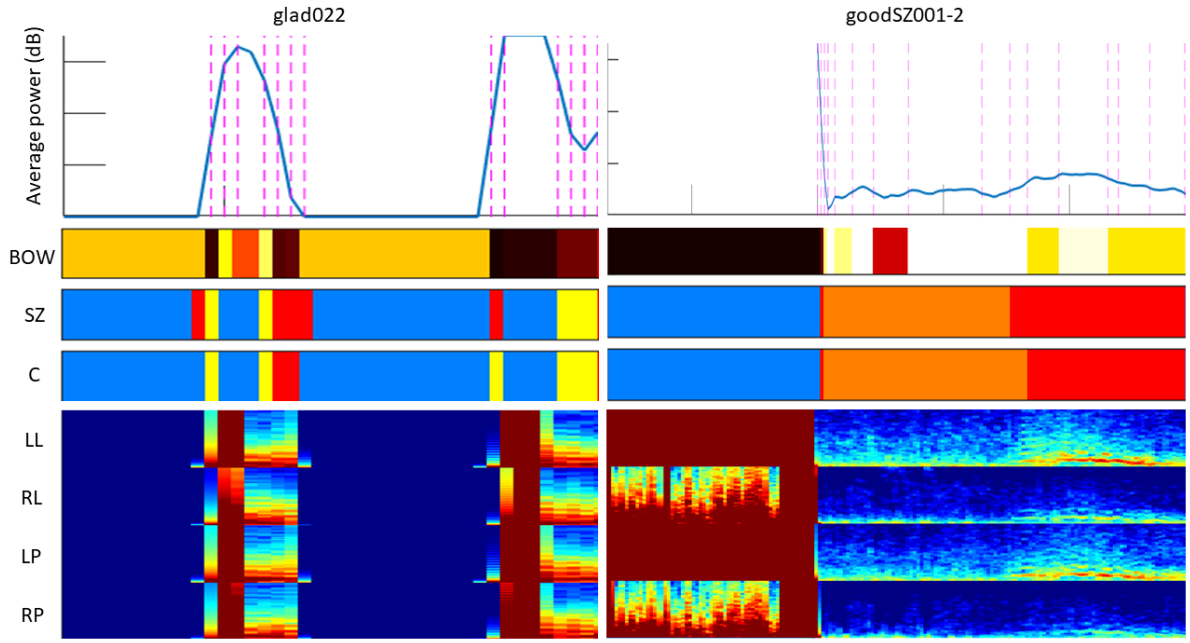


Figure 3.5: glad022 and goodSZ001: Bag of Words clusters (BOW), new labels (SZ, C), Average power (dB), for brain regions (LL, RL, LP, RP)

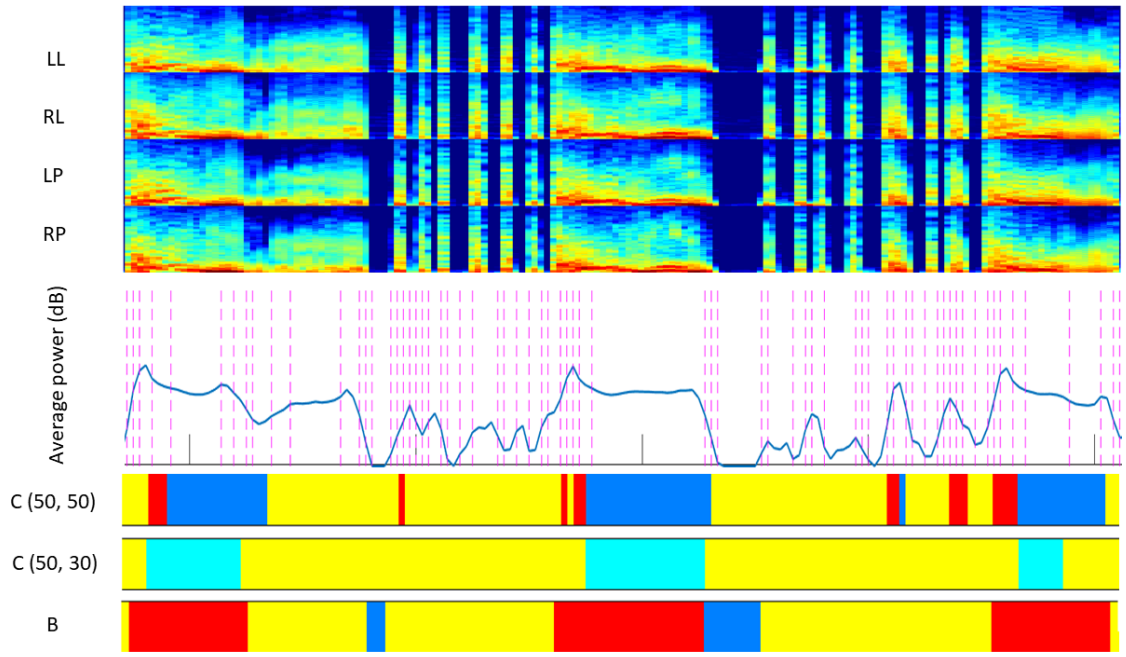


Figure 3.6: Bag of Words and Change Point Detection on glad003 under different settings for number of clusters (50 and 30)

3.6. Discussion

The results showed that using the frequency content of the signal can improve the labels on segments where the frequency is extremely high or extremely low. The results did show that different parameters were necessary for some patients and that the method could not be applied with the same parameters to all cases. The Bag of Words and Change Point Detection method can not be universally applied to long EEG recordings without validation from an expert and parameter tuning. In some cases, the labels from the Bag of Words and Change Point Detection did not create an improvement and the labels had

to be ”saved” by choosing a combination of the labels from a certain expert and keep seizures from other experts. This method does hold the potential to improve labels in a much shorter period of time compared to manually relabelling every segment. The GUI is a tool that can be used to validate labels quickly and look at raw EEGs.

In order to attempt to create a dataset with better labels expert B tuned the number of words and number of cluster parameters and made decisions on one of the following six scenarios for all patients with these three new labels. For example, the result in figure 3.6 led to the decision to use scenario 3 which meant using the labels from M which in this case means using the labels from expert B.

1. Use label C only (flag : 0)
2. Use label C + Seizure from M = SZ (flag : 1)
3. Use label M only (flag : 2)
4. Use label expert B only (flag : 3)
5. Use label expert B + Seizure from M (flag : 4)
6. Use label C + Seizure from M + Seizure from expert A (flag : 5)

The parameters and decisions for the labels can be seen in table A.1 in the appendix. The labels after using the previously discussed methods and tuned by one of the experts resulted the final labels that were used in the transfer learning experiments and the post processing algorithms.

3.7. Limitations

Combining the expert labels this way to generate new labels has the limitation that it is tuned by one expert. This makes the labels biased to the preferences of the expert who is tuning the labels. This same principle applies to scenario selection to rescue the labels when the Bag of Words and Change Point Detection method fails to generate better labels after tuning. The expert assigning the flags of the scenarios could have a preference towards its own labels which would introduce a bias in the data. Because expert B was tuning the labels and creating the scenarios, two out of the six scenarios involved using the labels from expert B. Even if the labels from expert B are better the preferred scenario would be to have each expert’s opinion weigh equally in the process of tuning the hyperparameters and making decision rules to prevent the labels from skewing towards one expert.

Another limitation is that the new labels are skewed toward the frequency content of the signal because of the clustering of power spectral density values. Noise in the EEG signal can influence the power spectral density values which results in less accurate clustering results. The power spectral density values give insight into the frequency domain but by only clustering on power spectral density values, this method loses morphological, spatial, and temporal information. Temporal and morphological information are important to identify short lived events like spikes, bursts, or artifacts and identifying rhythms and oscillations. Spatial information gives an understanding of the spatial distribution of brain activity that is useful for detecting the localization of the patterns in certain areas of the brain.

4

Transfer learning with Dense Convolutional Neural Networks

The previous chapter served as a step to improve and decide on the final labels. In order to deal with the limited data availability problem and leverage the information learned by Sparcnet 1, transfer learning experiments were done using the new labels. This chapter explored the potential of Transfer Learning on the noisier labels in order to classify the IIIC patterns. A total of 7 experiments were done with the around 12 hour long EEG recordings that were labelled by three experts and cleaned with the Bag of Words and Change Point Detection Models. These experiments were evaluated with machine learning evaluation metrics that give insight in the performance of the models on the classification task.

4.1. Data

The data was sampled from 140 patients who were in the ICU of the Massachusetts General Hospital. There were 71 males and 64 females. The average age was 49.49 years and the average length of the EEG recording was 13.37 hours. The measurements were in micro volts and recorded with an international 10-20 system at a sampling rate of 200 Hz. The full table 1 can be seen in appendix A.2.

4.2. Data pre-processing

In order to prepare the data for training the DenseNet CNN a number of steps were taken. The processing starts by taking the mono-polar EEG data and transforming it to L-bipolar and dropping the two midlines. This means each channel measures the voltage difference between two adjacent nodes. The L-bipolar montage is less prone to artifacts and best appreciate focal potentials. The data is organized into specific groups: left lateral, right lateral, left para-central, right para-central. This is done to improve the signal to noise ratio and put more emphasis on local differences. Following this conversion to bipolar, a notchfilter is applied to the signal with the aim of removing the power line noise that is present at 60 Hz. After removing the power line noise, a bandpass filter is used that keeps frequencies between 0.5 and 40 Hz. The bandpass filter ensures that high frequency noise and low frequency drifts are eliminated. In order to mitigate some artifact problems, the amplitudes of the signals are clipped between -300 to 300 microvolts. This clipping ensures that the data falls in the physiological range for EEG signals. The processed data is then segmented in overlapping windows. Each window is 10s long and the windows overlap 4s with the previous sample and 4s with the upcoming sample. These choices were made because this way the input of the data is the same way as in the original Sparcnet 1 model. This ensures that the EEG recording is divided in 10s segments with a moving window of 2s and creates the 8s overlap.

There is a high imbalance in classes in the dataset and the data comes from recordings of specific patients. With the aim of truly testing the model's generalizability across patients, the same patient data can not be present in both the training and test splits. This is why a stratified split is used that makes sure that each class is represented in the training, validation and test spit and that there is no overlap in patient data. The stratified split works by first initializing a class count dictionary per patient and a

global class order from the smallest class count to the largest.

Based on the class with the smallest amount of labels, patients are assigned to the training validation and test set in iterations. In each iteration, the current ratio is compared to a target ratio R_{target} (0.7, 0.15, 0.15) and patients are assigned to splits that are furthest away from this ratio until all the patients are in the splits. This way of spreading the patients generated the splits in tables 4.1 and 4.2 that were used to train and test the models.

Class	Training Set	Validation Set	Test Set
Other	1,550,739	370,679	336,829
Seizure	32,087	14,322	10,826
LPD	285,114	59,998	65,072
GPD	177,358	43,812	37,340
LRDA	243,121	20,592	18,817
GRDA	68,176	3,766	5,418

Table 4.1: Class Distribution across Training, Validation, and Test Sets in number of patients

Class	Training Set (%)	Validation Set (%)	Test Set (%)
Other	71.25	17.04	15.46
Seizure	61.18	27.33	11.49
LPD	67.39	14.18	18.42
GPD	65.43	16.15	13.13
LRDA	80.33	6.81	10.15
GRDA	86.19	4.76	8.05

Table 4.2: Class Distribution Across Training, Validation, and Test Sets

4.3. Loss function

In order to deal with the fact that the dataset is imbalanced, the weighted KL divergence loss function is used that allows differential weighting of classes. This function assigns higher weights to underrepresented classes which causes the model to pay more attention to this class. The loss function is given by

$$\mathrm{KL}_{\text{weighted}} = \frac{1}{K} \sum_{k=1}^{K} \sum_{m=1}^{M} w_m \left[\tilde{L}_{k,m} \cdot \log \frac{\tilde{L}_{k,m}}{L_{k,m}} \right] \tag{4.1}$$

The total number of samples that are present in the dataset and used in the weighted KL Divergence loss function is denoted by K. Each one of these samples gives a contribution to the overall loss. We make a distinction between the predicted probability distribution for each sample and the real probability distribution. The predicted probability distribution is denoted by $\tilde{L}_k$ and consists of the logits that are produced by the DenseNet CNN model transformed by a softmax function in order to create probabilities. The true probabilities L_k are one hot encoded vector where the real class has a value of 1 and the other classes have a value of 0. This vector can be seen as the ground truth which is compared against the model's predictions. Since its weighted KL divergence loss, the function has a weight vector w_m that penalizes the model for making mistakes on the under sampled classes. Because the model trains on batches of data the weighted KL divergence loss is averaged over all K samples. The reason for doing this is that by averaging over the samples the loss is normalized. By normalizing, the batch size does not have influence on the gradients and ensures stable updates while training.

4.4. Computational setup and distributed training

Computations were realized on a Linux system with a AMD Ryzen Threadripper 3960X 24-Core Processor, 251 GB RAM, Nvidia RTX A6000 GPU, and two Nvidia Quadro RTX 6000/8000 GPUs. Because of limited fast SSD memory and an HDD that caused program shutdowns, the deep learning experiments were continued inside a Docker container on another Linux system with an Intel(R) Xeon(R) W-2145 CPU @ 3.70GHz, 251 GB RAM and two Nvidia GeForce RTX 2080 Ti GPUs. The open source DL-framework Pytorch was used to construct the DenseNet CNN and load in the pretrained weights from Sparcnet. Because the data could not fit into memory (800 GB) dataloader and dataset classes from the Pytorch framework had to be constructed to load raw data from a Hierarchical Data Format (HDF5). Also because of the size of the dataset and the number of parameters from the model multiple GPUs were used in combination with the data parallel module available from Pytorch in order to greatly speed up training.

4.5. Hyper-parameters and experiments

With this basic set-up, 7 experiments were conducted, as summarized in table 4.3. We tested one main type of optimization algorithm, the ADAM optimizer. This optimizer shows better performance than the standard Stochastic Gradient Descent algorithm in most cases. Because of limited computing resources, a grid search across all parameters was not possible and the models were trained with two different batch sizes: 128 and 256 (experiment 1 & 4). Batch size 256 was chosen to continue with for the rest of the experiments because of the reduction in computing time, allowing the rest of the experiments to be done faster. Also limited by computing power, one experiment was done with the shuffle parameter that shuffles the data at every epoch for better generalization and to prevent the model to learn the order of the data (experiment 2). This parameter drastically increased training time. Because of training loss getting stuck in a plateau an experiment has been done with a larger learning rate to see if the solution ended in a local minimum (experiment 3). Furthermore, a number of transfer learning experiments were done. At first, only the classifier was retrained but three experiments were added. These experiments consisted of only retraining the classifier and the last denseblock (experiment 5), retraining the full model (experiment 6), and not loading any weights from Sparcnet (experiment 8). Finally, the old training data was added to the training set to see if retraining on more data would yield better results (experiment 7). The training used early stopping by monitoring the validation loss and stopping model training when the model does not improve within 5 epochs.

Experiment	Batch Size	Shuffle	Learning Rate	Unfreezed	Optimizer	Old Data
Exp 1	128	No	6.25*1e-5	Classifier	Adam	No
Exp 2	128	Yes	6.25*1e-5	Classifier	Adam	No
Exp 3	128	No	0.1	Classifier	Adam	No
Exp 4	256	No	6.25*1e-5	Classifier	Adam	No
Exp 5	256	No	0.6.25*1e-5	Dense Block 7	Adam	No
Exp 6	256	No	6.25*1e-5	Full	Adam	No
Exp 7	256	No	6.25*1e-5	Classifier	Adam	Yes
Exp 8	256	No	6.25*1e-5	-	Adam	No

Table 4.3: Transfer Learning Experiments

4.6. Evaluation metrics

Models were quantitatively evaluated using Accuracy, Confusion Matrices, Receiver Operating Curves (ROC), Area Under the Curve (AUC), Precision Recall curves (PR), F1 scores, Matthew's Correlation Coefficient, and Calibration plots. At the basis of classification, evaluation metrics stand the number of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) which can be seen in the Confusion Matrix in figure (4.1) The confusion matrix is made of the model's right classifications (TP & TN) and the model's misclassifications (FP & FN based on the actual classifications and the models predictions. The accuracy of the model can be defined by dividing the right classifications by all types of classifications in the confusion matrix summed over all classes as seen in equation 4.2.

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} TN_i}{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} TN_i + \sum_{i=1}^{N} FP_i + \sum_{i=1}^{N} FN_i} \tag{4.2}$$

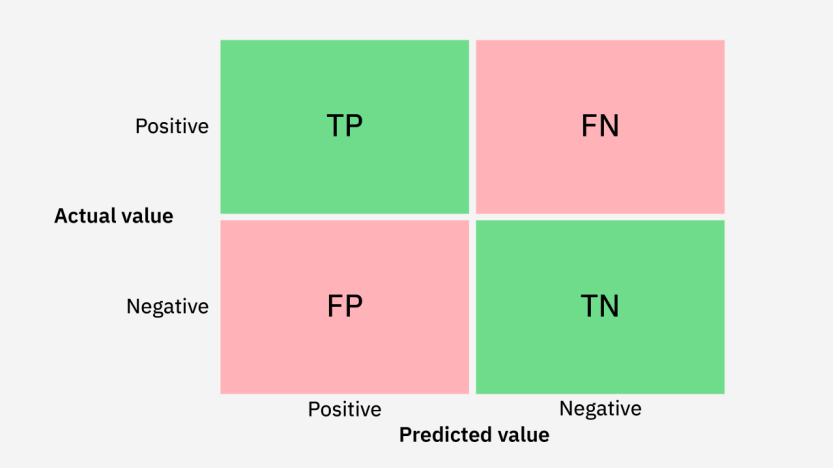


Figure 4.1: Confusion Matrix

ROC curves are graphical tools that can be used to evaluate the performance of binary classification models. In this case, binary classification means the classification of one of the IIIC classes versus the classification of anything other than that class. This means the presence or absence of a certain harmful brain activity pattern. It uses sensitivity on its y-axis and 1-specificity on the x-axis. Sensitivity is defined by the number of true positive (TP) cases that the model predicts divided by the number of true positive cases summed with the number of false negative (FN) cases. Also called the true positive rate (TPR), sensitivity is a measure of how many of the actual positive cases the model was able to capture. If you look at seizures, it is a proportion of how many of the actual seizures were classified correctly. If the model predicts 90 out of 100 seizures the sensitivity is 90%. Specificity, on the other hand, is calculated by the number of true negative (TN) cases divided by the sum of the true negative cases and the number of false positive (FP) cases. This means that if out of the 100 non seizures it correctly classifies 90 as non seizure the specificity is 90%. 1 - specificity is equal to the false positive rate (FPR) because FPR and specificity are complementary; the sum of true negative cases and false positive cases is a representation of the total number of negative cases. FPR is calculated by the number of false positive cases divided by the number of true negative cases and the number of false positive cases. The FPR is a measure of how many false alarms the classifier makes, which is important in the medical setting because of the valuable time of doctors. If the model out of the non seizures classifies 10 of them as seizures the FPR is 10%. The curve is created by evaluating, for a number of thresholds, what the TPR and FPR are. These thresholds can be seen as a decision boundary for which to classify the pattern based on the probability output of the model.

Just like the ROC curve, the precision-recall (PR) curve is also used to evaluate binary classifiers and is often used on imbalanced sets. It has precision on the y-axis and recall, which is the same as sensitivity, on the x-axis. Precision is a measure of the actual positive predictions that belong to the positive class and is calculated by dividing the number of true positive cases by the sum of true positive cases and false positive cases. This means that if the model predicts 100 seizures and 85 of them were actual seizures the precision is 85%. The PR curve focuses more on the classification of positive instances, which means a large number of negative samples (the other classes) will not skew the evaluation. For both curves, a higher area under the curve (AUC) indicates better performance and can be interpreted as a measure of the classifier's ability to distinguish between the positive and negative classes. To underscore why these metrics are important in the case of the unbalanced dataset consider the following example. There is a dataset with 10000 patients where 100 patients have a rare disease and the remaining do not. There is a model that detects 80 true positives 100 false positives, 9800 true negatives, and 20 false negatives. The accuracy is calculated as the ratio of all correctly predicted instances to the total instances: $\frac{80+9,800}{80+100+9,800+20} = \frac{9,880}{10,000} = 0.988$, or 98.8%. The accuracy metric is

actually misleading because of the imbalance in the data. The precision is: $\frac{80}{80+100} = \frac{80}{180} \approx 0.444$, or approximately 44.4%. The recall is $\frac{80}{80+20} = \frac{80}{100} = 0.8$, or 80%. This means that there are many false positives in the prediction of the model but it does capture most of the actual disease.

$$\text{Sensitivity} = \text{Recall} = \text{TPR} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4.3}$$

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}} \tag{4.4}$$

$$\text{FPR} = 1 - \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{4.5}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{4.6}$$

The F1 score is called the harmonic mean of precision and recall. The harmonic mean of precision and recall is two times the product of precision and recall divided by the sum of precision and recall. As seen in equation 4.7 you end up with the arithmetic mean of the false positives and the false negatives in the denominator. This means the F1 score equally considers false positives and false negatives. Because of the property that F1 is sensitive to both errors, this metric is especially useful for this dataset since it is highly imbalanced. In the example from earlier the F1 score is $2 \cdot \frac{0.444 \cdot 0.8}{0.444+0.8} \approx 2 \cdot \frac{0.3552}{1.244} \approx 0.571$, or approximately 57.1% which gives a more balanced indication of a model's performance.

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{\text{True Positives}}{\text{True Positives} + \frac{1}{2}(\text{False Negatives} + \text{False Positives})} \tag{4.7}$$

The MCC evaluation metric is also a more balanced metric than accuracy. It is a macro metric which means that the TN, TP, FN, and FP are calculated across all classes and their sums are used in equation 4.8. The denominator of the MCC equation is a normalization constant that makes sure that the value is between -1 and 1. In the extreme case of anti diagonal confusion matrix which means the true positives and true positives are zero, the value goes to -1. In the extreme case of the diagonal confusion matrix where the true positives and true negatives are 1 the MCC goes to 1.

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.8}$$

As seen in figure 4.2 by moving the decision threshold over output probability on the left the lines of the scores are generated on the right. The accuracy reaches a constant and is really high for a larger threshold while the recall goes down. This is an example of the precision-recall trade off and the accuracy being skewed just as in the calculation example from before. Other metrics that are used to evaluate the models are the macro variants of precision, recall, and F1 (equations 4.9, 4.10 and 4.11). The macro metrics offer a robust way of dealing with the class imbalance and representing false positives and false negatives across all classes.

$$\text{Macro Precision} = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_i}{TP_i + FP_i} \tag{4.9}$$

$$\text{Macro Recall} = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_i}{TP_i + FN_i} \tag{4.10}$$

$$\text{Macro F1} = \frac{1}{N} \sum_{i=1}^{N} 2 \times \frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{4.11}$$

In addition to quantitative evaluation, a qualitative evaluation was done using swimmer plots and spectrograms. A swimmer plot is a graphical way of showing the models' outputs over time per patient. The swimmer plots show the outputs from Sparcnet, the best performing model from the experiments, the labels from the experts, the original cleaned labels, the new bag of words labels, the labels after the boosting cascade, and the multinomial logistic regression model. The spectrograms are plotted above the swimmer plots in order to identify clear artifacts and changes in power over the frequency bins.

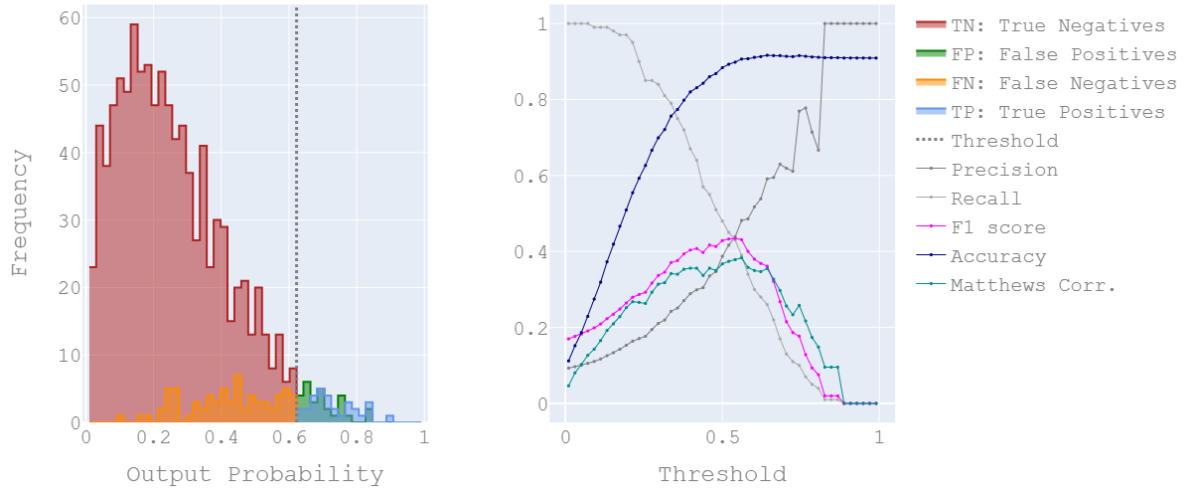


Figure 4.2: MCC and F1 compared to accuracy over thresholds

4.7. Results

4.7.1. Performance on ROC curves

The ROC curves from the experiments seen in figure 4.3 give insight in how well the models are able to discriminate between a class and the rest of the classes. The performance of the model is summarized in the AUC metric as described in section 4.6. This figure shows the performance of Sparcnet 1 compared to all other models that were trained in the transfer learning experiments. It can be seen that the AUC scores on the ROC curve from the Sparcnet model are 0.73 for Other, 0.77 for Seizure, 0.84 for LPD, 0.74 for GPD, 0.89 for LRDA, and 0.55 for GRDA. Notably, the low score for GRDA indicates that Sparcnet is barely better than random guessing for this class. Classes such as LPDs and LRDAs are classified reasonably well, while Other, Seizure, and GPD are classified moderately well.

Experiment 3 showed the lowest AUC across the board, with the exception of the LPD class, and had the lowest AUC of any class. This experiment, which had a higher learning rate and retrained the classifier, performed worse than guessing on Seizure, LRDA, and GRDA because of the AUC being lower than 0.5, which indicates systematic errors in the predictions. **Experiments 8 and 6** where no transfer learning had been done and where the entire model was retrained after weight initialization, had the lowest AUC for almost all classes. In Experiment 8, the AUC is actually lower than 0.5 for GPD, indicating that the model systematically makes wrong predictions. The same is true for Experiment 6 and the GRDA class. **Experiment 5** which involved unfreezing the last DenseBlock from Sparcnet, had similar AUCs as Sparcnet with the exception of the LRDA class. This model had the highest AUC of all models over all classes at 0.95 for LRDA. **Experiment 2** improved on Sparcnet AUC in every class except the Other class and the lateralized classes and had the highest AUC for seizure. **Experiment 7** which included the old training data from Sparcnet had higher AUCs than Sparcnet with the exception of the LRDA class. **Experiments 1 and 4** showed comparable or better performance on AUC than Sparcnet 1. These experiments involved retraining only the classifier but with different batch sizes. There is a noticeable increase in AUC for the GRDA class as it goes from 0.55 to 0.67. Also the AUC for the other class goes from 0.73 to 0.82 which is a big jump. Both experiments have a bit smaller AUC on the LPD class and experiment 4 is a bit better than experiment 1 on GPD and Seizure while the AUC for experiment 1 is a bit higher on LRDA.

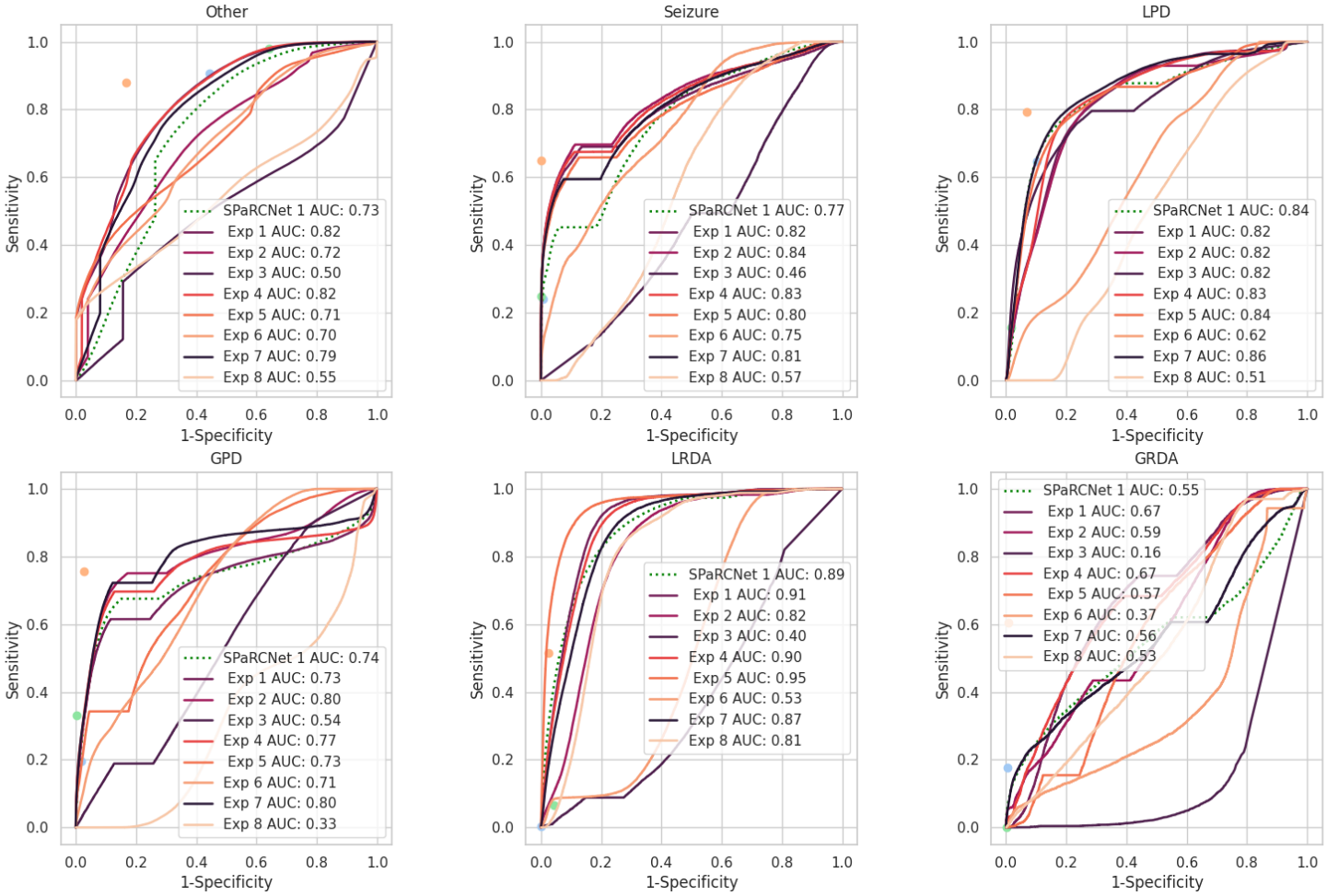


Figure 4.3: ROC curves for transfer learning experiments

4.7.2. Performance on PR curves

Figure 4.4 shows the precision recall curves of Sparcnet 1 compared to the transfer learning experiments. The curves give insight in the balance between precision and recall and are summarized in the AUC metric described in section 4.6. It can be seen that Sparcnet's AUC scores for the PR curve are 0.82 for Other, 0.25 for Seizure, 0.45 for LPD, 0.37 for GPD, 0.26 for LRDA and 0.04 for GRDA.

The low performance of precision and recall for the GRDA can be seen across the board in all the experiments. None of the AUCs surpass the highest score from Sparcnet of 0.04. **Experiments 1 and 4** show big improvements in Other and Seizure classes, moderate improvement in the GPD class and lower AUC in the LPD and LRDA classes. **Experiments 8 and 6** again rank in the top lowest scoring models based on the AUC in the PR curves. Experiment 8 scores consistently lower on AUC on all classes. Experiment 6 scores the same with the exception of the Other class. Except for experiment 3 all models had a higher AUC for "Other" indicating better balance between precision and recall for the different thresholds on that class. Also for seizures most experiments had a clear better precision and recall balance. The experiments showed around the same or worse balance between precision and recall on the "LPD" and "GPD" classes in comparison to Sparcnet 1. **Experiment 5** was the only one to surpass Sparcnet 1 on the "LRDA" class and the rest of the experiments did not perform better.

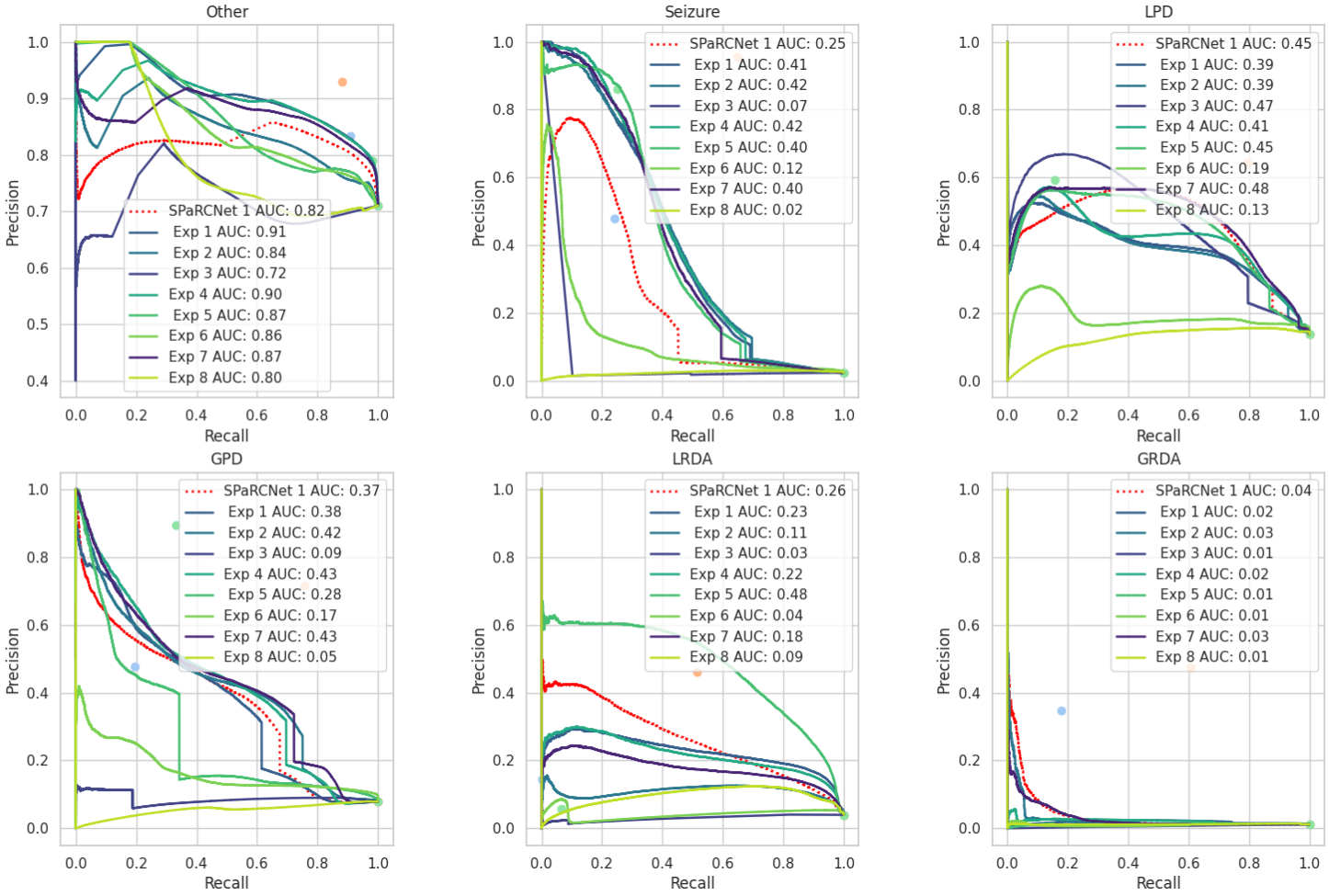


Figure 4.4: PR curves for transfer learning experiments

4.7.3. Evaluation metrics

The macro metrics and class specific metrics in tables 4.4, 4.5, 4.6 and 4.7 are calculated by forcing the model to make a decision based on the probabilities of the model. The entire array of probabilities is passed to a function that returns the index with the largest number. The original Sparcnet model outperforms any of the experiments on accuracy, Macro F1 and MCC. The accuracy is considerably higher with 11% more than the second best model of experiment 4 on this metric. The higher accuracy makes sense because the class specific metrics in table 4.5 show that Sparcnet has the highest F1 (0.77) for the Other class which by far the largest class in the dataset. Sparcnet does not have the best metrics on Macro Precision and Recall. **Experiments 1 and 4** both showed better macro precision and recall but because macro F1 and MCC take all classes into account these experiments don't score better on the balanced metrics. There is a clear decline in the macro metrics for experiments 2, 3, 5 and 8. This decline is accounted for by the class specific results. **Experiments 2, 3 and 5** show zero precision, recall and F1 on the other class. Experiment 3 shows zero precision, recall and F1 on GPD. **Experiment 8** does the same for LPD, GPD, and GRDA but has a precision for the other class of 1 which indicates over classifying Other. The models where the class specific metrics are zero across the board suggest that these models do not classify that specific class at all and are the reason for the low macro evaluation metrics. **Experiment 6** is another example of showing a higher accuracy because of its performance on the Other class but scoring really low in the rest of the macro metrics. **Experiment 7** does not show the decline as experiments 2, 3, 5, 6 and 8 but is not better than experiments 1 and 4 in terms of macro metrics.

4.7.4. Confusion matrices

In the confusion matrix of figure 4.5 the actual labels and the predicted labels of Sparcnet 1 can be seen. Sparcnet 1 is best at predicting the ”Other” class as 71% of actual labels were predicted as such by Sparcnet 1. The ”LPD” class is second best and has 57% of actual labels predicted while being confused in 27% of cases with the ”Other class. For the other classes, the predicted labels belonged to less than 50% of the actual labels. The ”Seizure” and ”LRDA” classes ended up with 48 and 47% of the actual labels. The ”Seizure” class was mostly confused with the ”LPD” class and the ”LRDA” class was confused in 41% of the cases with the ”Other” class. The model had the lowest actual predictions for the ”GPD” and ”GRDA” classes with 27% and 21% of the actual labels. The ”GRDA” class was mostly confused with the ”Seizure” class and the ”GPD” class with the ”Other” class.

The confusion matrix in figure 4.6 shows the actual labels and predictions of the retrained model with a batch size of 256. The new model best captures the class is ”LRDA”. For this class 84% of the actual labels were classified by the model. The ”GRDA” class had an increase of 12% and the ”Seizure” class of 1% in the classification of the actual labels. These two improvements went at the cost of the correct detection of the other classes. The ”LPD”, ”GPD” and ”Other” classes lost 11%, 7% and 25% of the correctly predicted classes. The confusion of the ”Other” class with the rest of the classes is less apparent but 43% of the actual GPDs were confused as LPDs and there is an increase in the confusion of other classes with the ”LRDA” class.

4.7.5. Swimmer plots

Figure 4.7 shows the swimmer plots of patients ”glad004” and ”glad017”. These two examples from the test data show an improvement over Sparcnet 1 in glad017 and an increase in misclassification due to the bias towards ”LRDA” in glad004. Sparcnet 1 is clearly better at predicting LPDs in glad004, which are both present in the new labels from the Bag of Words model and the labels from expert B. The retrained model from experiment 4 which had batch size 256 is over-classifying the ”LRDA” class as can be seen by the green TF 256 bar. In glad017 the retrained model seems to follow the new labels more closely and classifies the samples correctly whereas Sparcnet 1 is wrongly classifying segments as the ”GPD” class.

Experiment	Accuracy	Macro Precision	Macro Recall	Macro F1	MCC
Sparcnet 1	**0.6369**	0.3884	0.4526	**0.3848**	**0.3425**
Exp 1	0.4691	0.4136	0.4559	0.3065	0.2650
Exp 2	0.1019	0.2673	0.3755	0.2290	0.1418
Exp 3	0.1053	0.0720	0.2078	0.0907	0.1181
Exp 4	0.5244	**0.4150**	**0.4823**	0.3575	0.2980
Exp 5	0.0930	0.2877	0.3401	0.1436	0.1077
Exp 6	0.3130	0.1863	0.1987	0.1408	0.0946
Exp 7	0.4247	0.3774	0.4491	0.3226	0.2584
Exp 8	0.1717	0.1919	0.2989	0.0934	0.1217

Table 4.4: Macro Metrics for Transfer Learning Experiments

Experiment	Other PR	Other RC	Other F1	Seizure PR	Seizure RC	Seizure F1
Sparcnet 1	0.8410	0.7112	0.7706	0.1587	0.4831	0.2389
Exp 1	0.8946	0.4902	0.6334	0.3084	0.4992	0.3812
Exp 2	0.0000	0.0000	0.0000	0.5017	0.3875	0.4373
Exp 3	0.0000	0.0000	0.0000	0.0194	0.5687	0.0376
Exp 4	0.8883	0.5581	0.6855	0.3283	0.4929	0.3941
Exp 5	0.0000	0.0000	0.0000	0.2101	0.5515	0.3043
Exp 6	0.8878	0.3819	0.5340	0.0600	0.4024	0.1044
Exp 7	0.8906	0.4181	0.5691	0.1402	0.6261	0.2291
Exp 8	1.0000	0.1703	0.2911	0.0302	0.8115	0.0583

Table 4.5: Metrics for Other and Seizure Classes

Experiment	LPD PR	LPD RC	LPD F1	GPD PR	GPD RC	GPD F1
Sparcnet 1	0.5355	0.5711	0.5527	0.5036	0.2749	0.3556
Exp 1	0.3520	0.4568	0.3976	0.7630	0.0764	0.1389
Exp 2	0.4460	0.2980	0.3573	0.5385	0.2933	0.3798
Exp 3	0.3970	0.6714	0.4989	0.0000	0.0000	0.0000
Exp 4	0.3759	0.4634	0.4151	0.7006	0.2044	0.3164
Exp 5	0.5364	0.2683	0.3577	0.9028	0.0286	0.0554
Exp 6	0.1398	0.1712	0.1539	0.0000	0.0000	0.0000
Exp 7	0.4869	0.4857	0.4863	0.5755	0.2868	0.3828
Exp 8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 4.6: Metrics for LPD and GPD Classes

Experiment	LRDA PR	LRDA RC	LRDA F1	GRDA PR	GRDA RC	GRDA F1
Sparcnet 1	0.2565	0.4656	0.3307	0.0349	0.2100	0.0599
Exp 1	0.1376	0.9563	0.2406	0.0261	0.2566	0.0474
Exp 2	0.1044	0.5138	0.1736	0.0133	0.7606	0.0262
Exp 3	0.0156	0.0049	0.0074	0.0002	0.0018	0.0004
Exp 4	0.1672	0.8406	0.2789	0.0299	0.3346	0.0548
Exp 5	0.0689	0.9817	0.1288	0.0079	0.2108	0.0153
Exp 6	0.0260	0.2277	0.0467	0.0040	0.0090	0.0055
Exp 7	0.1549	0.5030	0.2369	0.0165	0.3750	0.0316
Exp 8	0.1211	0.8114	0.2108	0.0000	0.0000	0.0000

Table 4.7: Metrics for LRDA and GRDA Classes

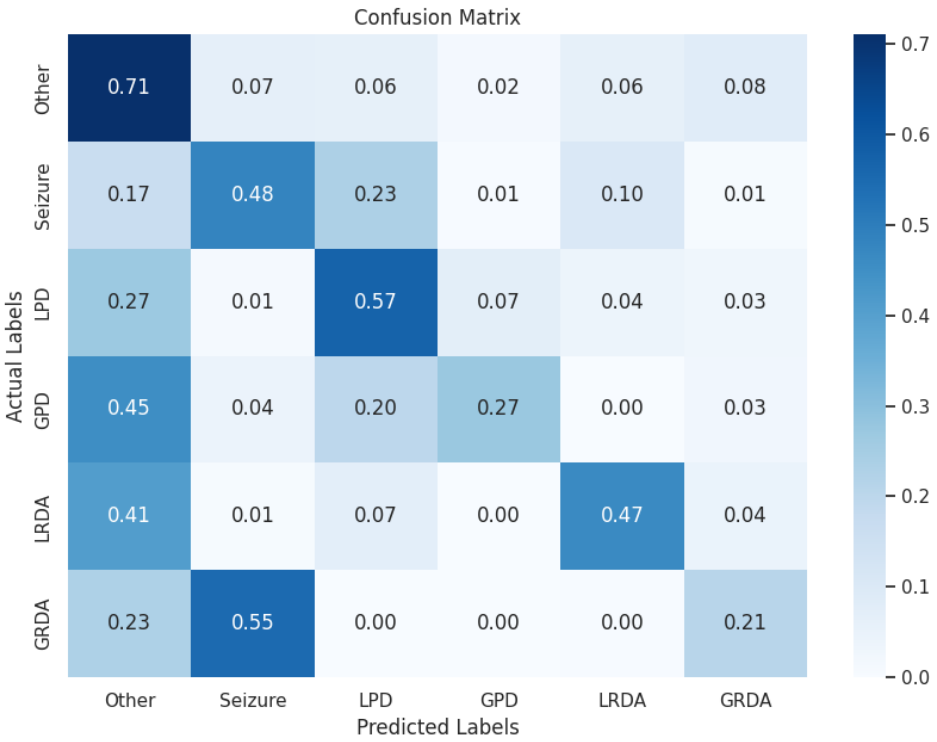


Figure 4.5: Confusion Matrix Sparcnet 1

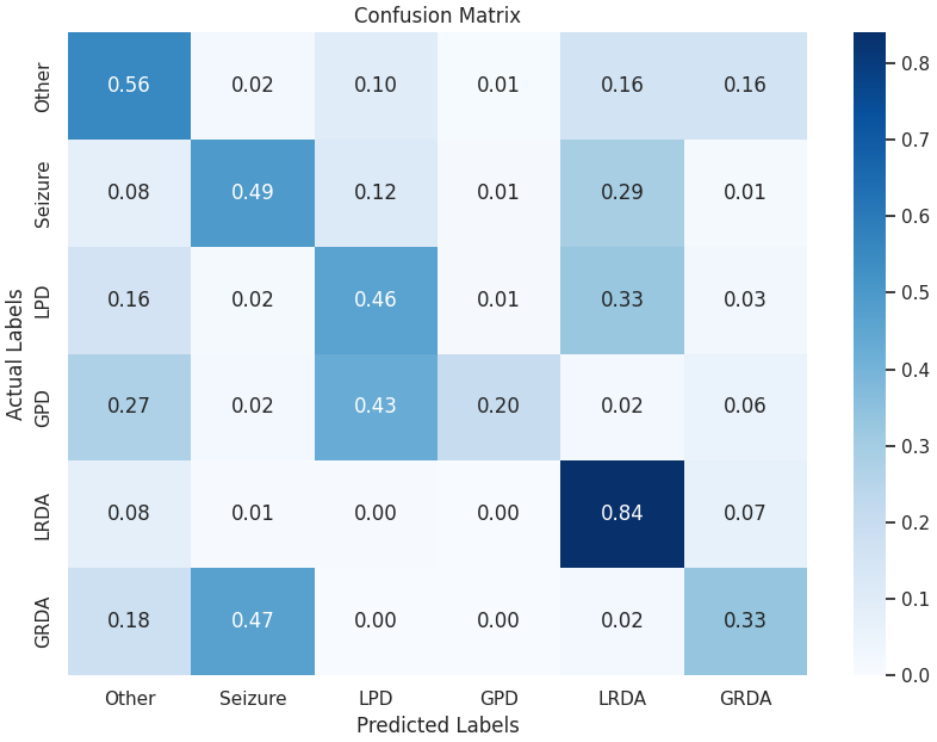


Figure 4.6: Confusion Matrix Exp 4

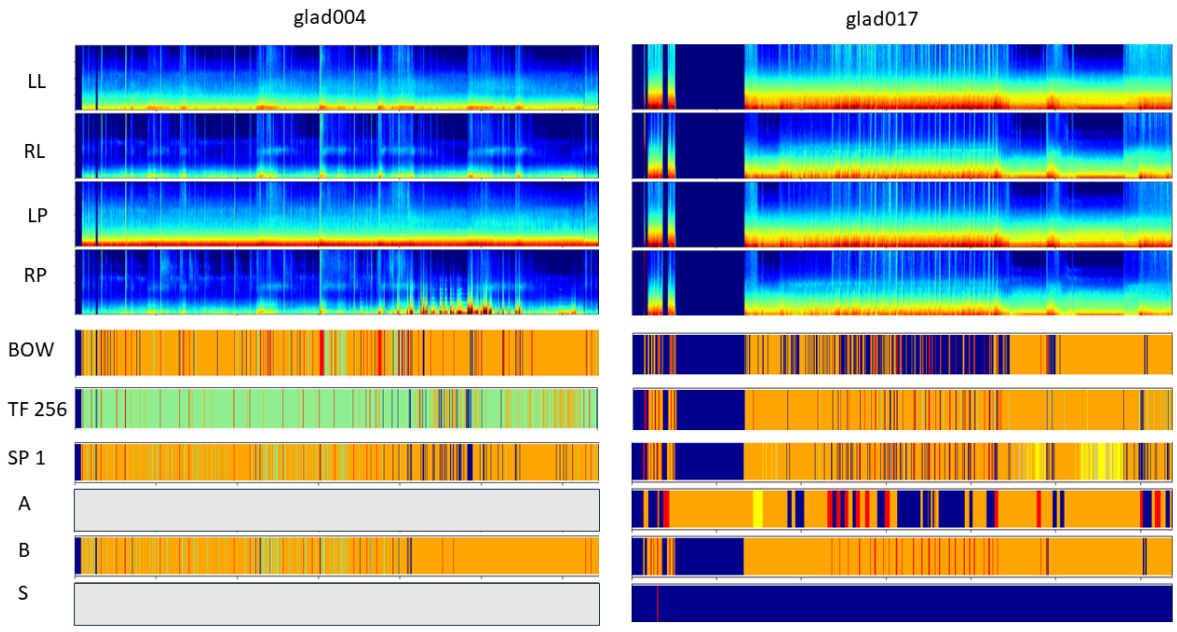


Figure 4.7: glad004 and glad017 transfer learning swimmer plots and spectrograms

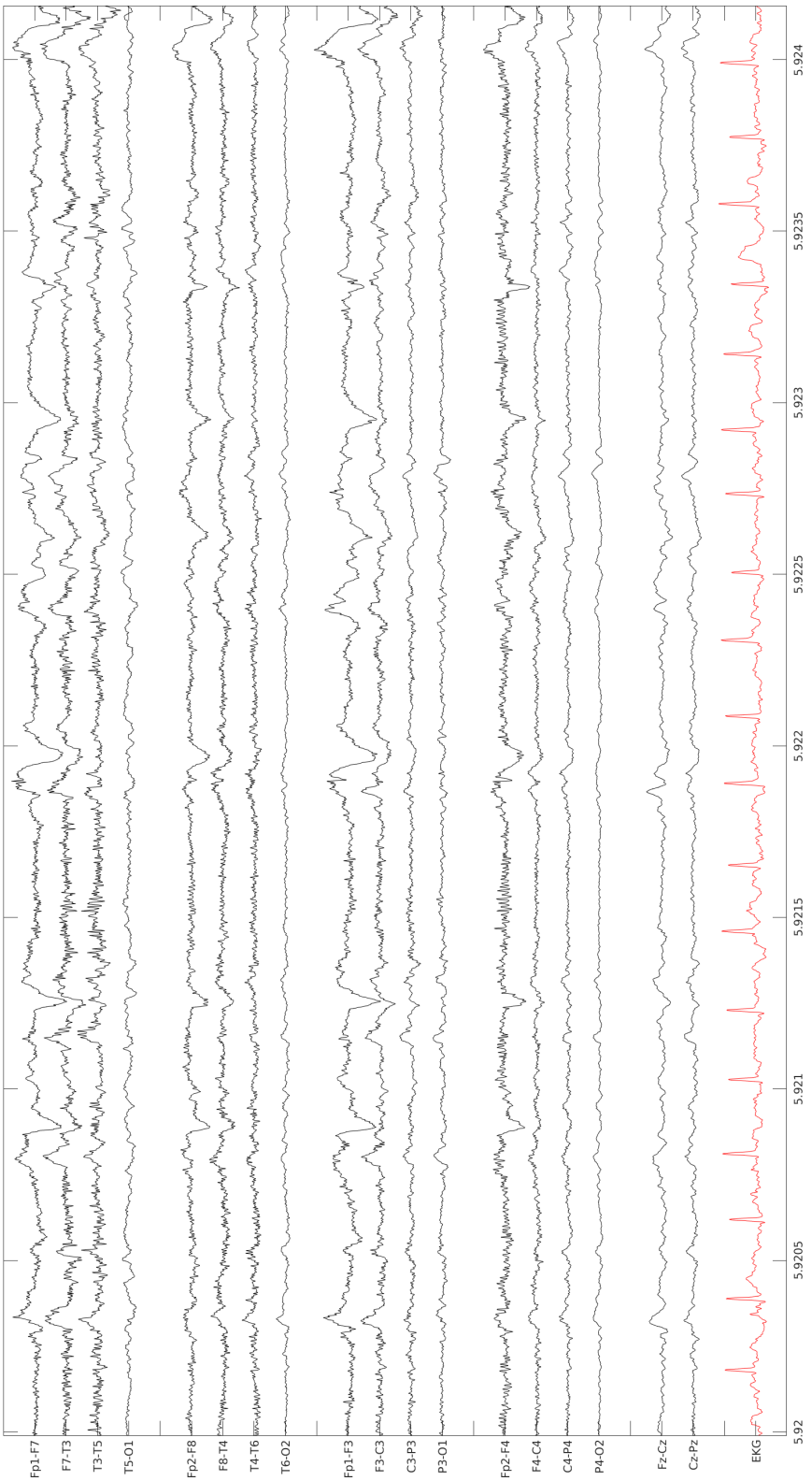


Figure 4.8: Raw EEG glad017

4.8. Discussion

The results from the transfer learning experiments show that unfreezing more layers and training more parameters does not help with capturing the patterns in the data. Training more parameters of the DenseNet Convolutional Neural Network on this data did not increase performance compared to just using Sparcnet 1. The results show how the large ”Other” class can skew accuracy. Retraining the entire network (experiment 5) has a higher accuracy than just retraining the last denseblock and the classifier (experiment 6) even though the balanced F1 and MCC metrics are better. This is caused by the fact that the denseblock model from experiment 5 is not predicting any ”Other” classes. This behavior can be seen across all experiments for at least one of the classes except for experiments 1, 4, and 7. Combining these results with the training loss plots in section A.4, it can be concluded that models trained without transfer learning and models where more parameters are trained are not learning effectively from the data and labels.

Experimenting with the batch size, shuffle, and learning rate hyper-parameters showed vastly different results in the evaluation metrics and training loss plots (training losses can be seen in section A.4). From these three hyper-parameters batch size had the least influence but still increased the macro evaluation metrics. Shuffling the data every epoch showed an initial increase in validation loss and a decreasing training loss that seems to plateau. Because shuffling the data at each epoch should theoretically increase generalizability this could be a sign that the labels are still too noisy. A much larger learning rate resulted in the lowest macro F1 score of the transfer learning experiments and an increasing training loss while training. This concludes that the learning rate was set way too high causing it to overshoot the optimal solution and should be closer to the original learning rate. It is clear that the hyperparameters have a large influence on the results.

The confusion matrix in figure 4.6 also showed that the best performing model of the transfer learning experiments is biased towards LRDA and only improves detection for Seizures and GRDA. The better detection of seizures and GRDA is attributed to the loss function and the larger weights for these classes which have the least samples in the dataset.

As the results show transfer learning with the current set of hyper-parameters and experiments trained on the labels from the Bag of Words and Change Point Detection method is not a solution to help capture the complex nature of ICU EEG data and their IIIC patterns. The results gave evidence for two possible reasons for the negative results in the transfer learning experiments. First It could be that the quality of the data is still not good enough and that there is too much noise for the model to learn anything. The limitations from the Bag of Words and Change Point Detection method could have influence on the transfer learning experiments. Second the hyper-parameters that have a large influence on the results and are not being set right. The learning rate, early stopping parameter, and shuffle parameter have a big effect on the evaluation metrics, training losses, and the way that the model learns.

The model’s inability to learn the patterns from the data could also be due to the combination of these two factors but there is an indication that the labels are not yet ready to be used to conduct deep learning experiments. This conclusion is supported by the results and limitations from chapter 3 but also by the swimmer plots, the evaluation metrics, and training plots seen in the results. The swimmer plots in figure 4.7 of some of the training patients showed high variability between experts and the new labels. In the case of glad017 the labels between expert A, B and S all vary substantially over the 12 hour recording. The labels of expert S were mostly incorrect, expert A labeled most segments as ”Other” around halfway of the EEG and expert B labeled these all as ”LPD”. The labels of expert A and S have influence on the Bag of Words pipeline and introduce more ”Other” classes around halfway of the EEG in the final labels. The raw EEG in figure 4.8 which is sampled from one of these points labeled as ”Other” shows that there are LPDs present in the sample. This is an indication that not all labels might be correct in the training, validation, and test data. In glad004’s case, experts A and S had no labels. The final labels ended up looking different than the labels from expert A, they introduced more ”LRDA”, ”Seizure” and ”Other” classes.

4.9. Limitations

As stated in the discussion the negative results from the Transfer Learning experiments could be due to inaccurate noisy labels or because of the decisions that were made for one of the many hyperparameters and in the training process of the DenseNet model. Certain decisions in the training process

and on the hyperparameters were made because of computing and time constraints but it is not certain that these are optimal. The training set was 600 GB and consisted of over 2 million 16x2000 samples. This means it can't be loaded into memory resulting in a lot of iterative reading operations from an HDD or SSD. If the storage disks are old and don't have fast memory then the iterative operation of data loading is slow. This can cause a bottleneck even with a good CPU and powerful GPUs. Even though the DenseNet CNN models in the Transfer Learning experiments had a thousand to a million parameters, which is relatively few parameters, the model still took a really long time to train because of the many denseblocks that are fully connected between layers and the large amount of training data. The computing and time constraints led to the following choices and limitations:

1. Early stopping with patience of 5 epochs on every experiment
2. No Shuffling of the batch at each epoch for all experiments
3. No adaptive learning rate
4. No K-Fold cross validation during training
5. No Gridsearch across the combination of hyperparameters

Because one epoch often took more than an hour early stopping with a patience parameter of 5 epochs was implemented. It could be the case that model training was ended prematurely because of this and that the models are getting stuck in a plateau and only starts learning a longer period of time.

Data shuffling, which is good practice, could not be done because it would greatly increase the time it took do to one epoch and would reduce the amount of experiments that could be done. The training loss plot shows interesting behavior that could indicate that the validation loss is decreasing after an initial increase but training is stopped because of the early stopping logic. By not shuffling the data for the other experiments training can become inefficient because the model sees the same patterns in a batch which makes it hard for the model to learn diverse features.

As seen from the results and the discussion of experiment 3 in the previous section increasing the learning rate resulted in the models not learning. Even though the larger leanring rate did not yield better results there is no guarantee that the old learning rate of Sparcnet is the best option and the models could still get stuck in a local minimum or overshoot the global optimum. The more extensive way of dealing with this is by using an adaptive learning rate that are known to overcome plateaus in the training loss and achieve smoother convergence [93]. Implementing this in the training loop would also reduce the need to do manual tuning or a gridsearch over a few learning rates.

Just like an adaptive learning rate implementing K-fold cross validation which would entail training and validating over multiple folds of the combined training and validation set could be an improvement of the current training loop. There are advantages to K-Fold cross validation like reducing variance, maximizing data use, improve model generalization and reduce overfitting [59].

Although the experiments are done for some settings like batch size and learning rate other hyperparamters were not tuned at all. There were no changes in optimization algorithms, different patience values for early stopping and model depth. A gridsearch to find the optimal combination of parameters could not be implemented because of computing power.

A big problem of this dataset is the imbalance in classes. This problem is currently solved by using the weighted Kullback-Leibler divergence loss function. The loss function is one of the most important aspects of training deep learning models. Using a different loss functions might give better results especially since the weighted loss function might not capture the complexity of the imbalanced dataset and its efficiency can vary based on dataset feature. Weighted Kullback-Leibler divergence loss is an algorithm level solution to the imbalanced dataset problem and data level solutions are not explored.

5

Feature based post processing

This chapter deals with the postprocessing methods that are applied on the output of Sparcnet. Both the Boosting Cascade and the Multinomial Logistic Regression model leveraged features generated from the raw EEG data in order to try to reduce the False Positive Rate of the classes that are associated with harmful brain activity: Seizure, LPD, GPD, LRDA, and GRDA. The models were quantitatively evaluated with the evaluation metrics from section 4.6 and qualitatively through swimmer plots.

5.1. Boosting Cascade

The process of using a boosting cascade to reject features can also be called a bad sample detection algorithm. The goal of this model is to reclassify samples that were unjustly classified as a ”non Other” class to ”Other”. Very clear artifacts that the SparcNet model classifies as harmful brain activity should be filtered out. This method is based on the bad channel detection from Dirks et al. who showed an improvement of the macro F1 score [15]. The pipeline begins with the calculation of features that hold information about the signal that can be used to detect these bad samples. These features include line length, Shannon entropy, zero crossing count, Non Linear Energy Operator (NLEO), Kurtosis total power, and total power for each of the frequency bands (α, β, θ, γ and δ). Line length is the total length of the waveform and is calculated by the absolute difference between each two following points $x[i+1]$ and $x[i]$ summed over all points in the sample (equation 5.1). The line length feature is a measure of the waveform dimensionality changes and is sensitive to the variation of the signal amplitude and frequency.

$$\text{Line Length} = \sum_{i=1}^{N-1} |x[i+1] - x[i]| \tag{5.1}$$

Shannon entropy is a feature that measures uncertainty. It is a non-linear metric that can quantify the degree of complexity in a time series and a higher entropy means the signal is more complex and less predictable [66]. To calculate the entropy of the signal a histogram of the signal is made. The values and their counts are put in 50 bins creating a probability density function. To estimate the probability distribution $p(x_i)$ the histogram is multiplied by the bin widths. This output is normalized and zero probabilities are removed before using the entropy equation 5.2.

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i) \tag{5.2}$$

Zero crossing is a feature that measures when a signal goes from positive to negative and the other way around (equation 5.3). The number of zero crossings can be associated with subtle epileptiform discharges and is able to discriminate between patients with epilepsy and normal subjects [68].

$$Z = \sum_{i=1}^{N-1} \mathbb{1}((x[i] \cdot x[i+1]) < 0) \tag{5.3}$$

Kurtosis is a well know statistical feature that represents the shape of the distribution tails compared to the overall shape of the distribution. Samples with high Kurtosis could indicate outliers and artifacts. The function that calculates this feature first checks if the signal is not zero otherwise it returns a zero. After this check the sample mean $\bar{x}$, standard deviation s, sample size n and the sample values x_i are used in the bias corrected Fisher Kurtosis (equation 5.4). This version of the Kurtosis formula makes sure the feature is unbiased and adjusted according to Fisher's definition. This ensures a more accurate measure of how the tails compare to the rest of the distribution.

$$K_{f,\text{unbiased}} = \frac{n(n+1)}{(n-1)(n-2)(n-3)} \left(\sum_{i=1}^{n} \frac{(x_i - \bar{x})^4}{s^4} \right) - \frac{3(n-1)^2}{(n-2)(n-3)} \tag{5.4}$$

As explained in section 2.1 the power of the frequency bands and their ratios hold important information about what is going on in the brain. The most useful frequency ranges are alpha, delta, and theta frequencies because they show the highest correlation with epilepsy and seizures. By using Multi-taper spectral estimation the power spectral density is computed of the signal. Multi-taper spectral estimation needs a number of parameters to compute the respective dB values for frequency bands over time. The first parameter is the moving window. In the calculation of the spectrogram, a moving window of 4 seconds with a 2 second step was used. The second parameter is the time bandwidth product and the number of tapers. The time bandwidth product is a parameter that controls the spectral ratio and the spectral leakage while the number of tapers is the amount of DPSS. The time product was set at 2 and three tapers were used. The third parameter is the frequency band pass which is the range of interest for the spectral analysis and was set at the range from 0.5 - 20 Hz. Finally, the last parameter is the sampling rate which is the sampling rate from the data at 200 Hz. From the power spectral density, a number of features are computed. By summing over all frequency bins the total power of the sample is generated. The same thing is done for the alpha, theta and delta frequency ranges creating total alpha power, total theta power and total delta power. Since their ratios hold information about harmful brain activity these features are also computed.

These features are computed for each of the 16 channels in the EEG samples. Because the goal is to reject an entire sample in the boosting cascade these features need to be transformed from channel specific features to sample specific features. This is done by using statistical features that capture information over the entire sample.

The statistical features that were calculated consisted of minimum and maximum values, the standard deviation, the median, the interquartile range and the confidence intervals. The minimum and maximum values give information of the range of values. The median is a central tendency measure that is more robust than the mean value. The standard deviation is a measure that gives information about the spread around the mean. The 2.5th, 25th, 75th and 97.5th percentile ranges give insight in the spreading of the labels and the interquartile range is calculated from the difference between the 75th and 25th which measures the spread of the middle 50% of the data. The confidence intervals help understand the tails of the data. All these features help understand the patterns and variability in the data.

Before training the Boosting Cascade all feature are standardized by subtracting the feature's mean and dividing by the feature standard deviation. The features are then loaded in with the labels and the Other class is changed to 0 and the rest of the classes to 1. For each feature the the Spearman correlation coefficient is calculated with respect to the labels. The sign of each feature is adjusted to ensure a positive correlation with the label. This is done because otherwise the direction of the threshold can be both ways for rejecting samples as artifacts. The sensitivity cutoff is initialized at 0.999 and the initial global false negatives and true negatives are set to zero. The features are then ranked based on their false positive rate for the sensitivity cutoff. The feature with the lowest false positive rate is selected and its threshold is calculated. All samples that have values below this threshold are thrown away, the feature is removed from the data and the steps are done again until all the features have been processed. At each step, the global sensitivity and false positive rate are tracked. Figure 5.1 shows the training steps of the cascade and the trade off between a worse sensitivity and a better false positive rate. Feature generation and training are both done on the same training set training set of section 4.2.

In order to improve generalizability, the number of features used in the cascade is validated on the validation set from section 4.2. In figure 5.2 the F1 score is plotted against the number of features. The

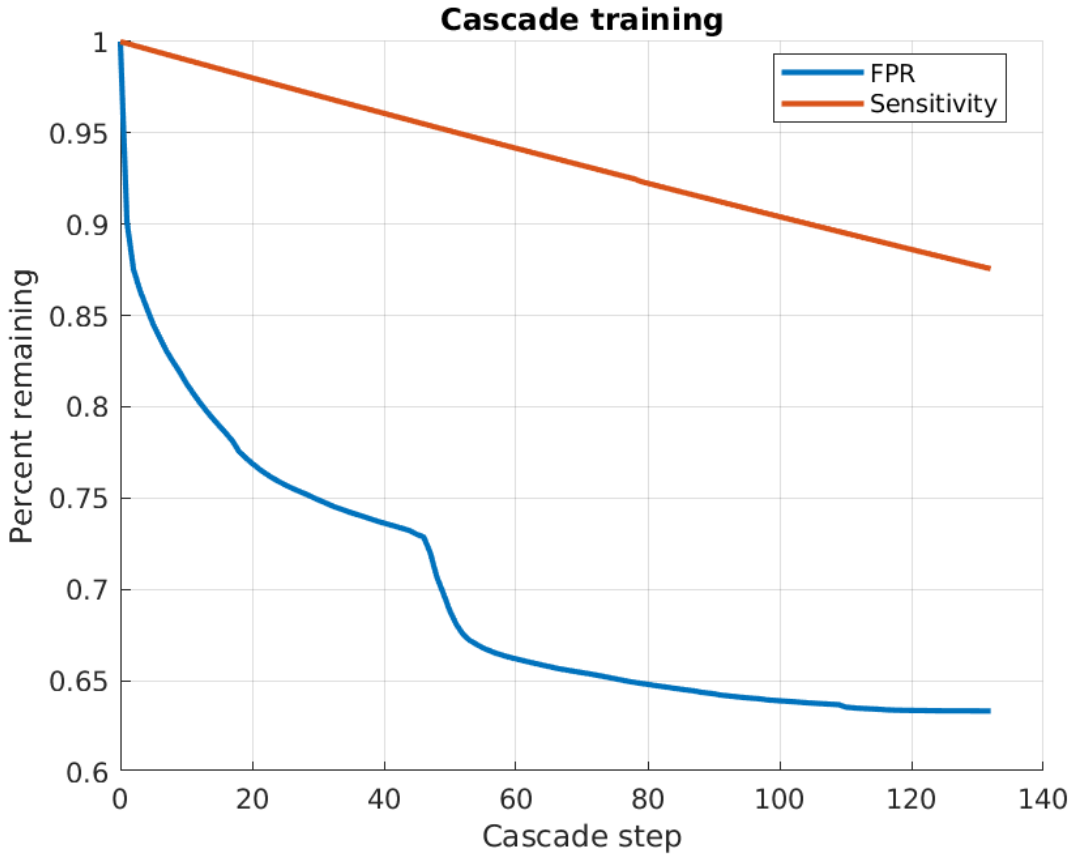


Figure 5.1: Training steps of the cascade

F1 score reached a plateau and the number of features that was used to evaluate the cascade on the test set which also stems from the splits of section 4.2 was 40.

5.2. Multinomial Logistic Regression

In order to further explore postprocessing and make use of the calculated features from the boosting cascade we fit a Multinomial Logistic Regression (MLR) model on a vector of the logits from Sparcnet concatenated with all the features from that sample. Multinomial Logistic Regression is a generalization of the binary Logistic regression classifier to a multiclass problem with more than two possible discrete outcomes. MLR is a probabilistic model of the categorically dependent variable given a set of independent variables. As seen in equation 5.5 the probability for class k is given by a linear combination of feature vector x and parameters β in a softmax function where it is being normalized to ensure that the sum of probabilities is equal to 1.

$$P(Y = k \mid \mathbf{x}) = \frac{e^{\beta_k^T \mathbf{x}}}{\sum_{j=1}^{K} e^{\beta_j^T \mathbf{x}}} \tag{5.5}$$

The fitting process consists of finding the right parameters β that maximize the likelihood function which is the probability density of the data as a function of the parameters. The likelihood in essence tells us how well the parameters fit the distribution of the data. Mathematically this is done through the product of the probabilities from equation 5.5 for each sample. Because this multiplication over a large dataset can cause a number to go to zero, the log-likelihood is taken which transforms the product into a sum and makes computations easier. The derivation from probability to log likelihood leads to equation 5.6 where the total log-likelihood is a sum of the log probabilities over all true classes. Because there is an imbalance in the data the classes have a weight w_{y_i} that corresponds to the frequency of the

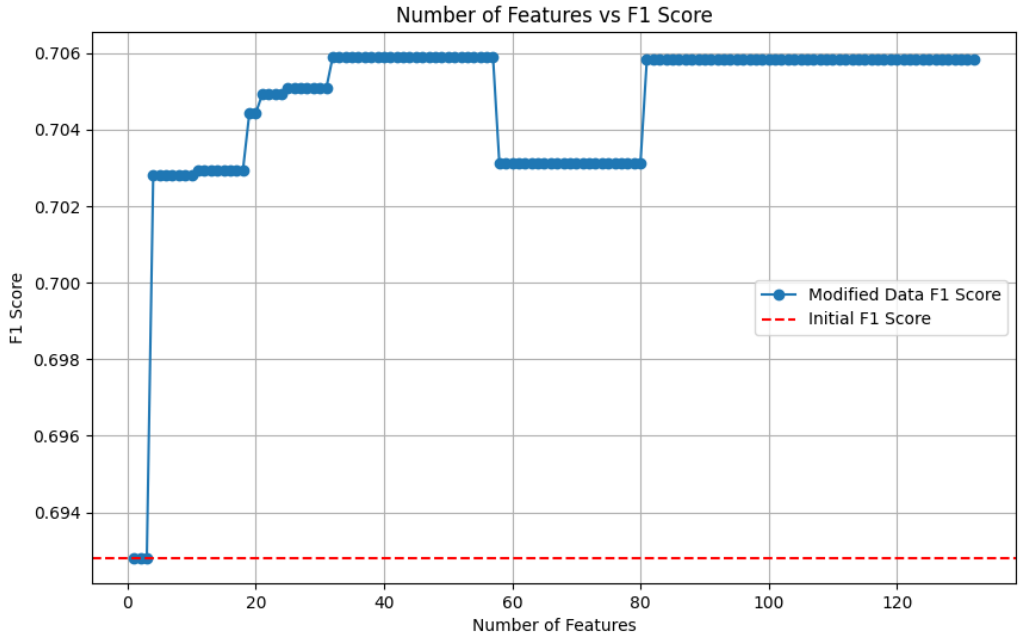


Figure 5.2: Validation cascade based on F1

class in the training set. Each term $\beta_{y_i}^T \mathbf{x}_i$ is the logit favouring class y_i and $\log\left(\sum_{j=1}^{K} e^{\beta_j^T \mathbf{x}_i}\right)$ is the normalization factor. The parameters of the MLR are calculated by maximizing the log-likelihood for the entire training set.

$$\ell(\beta) = \sum_{i=1}^{n} w_{y_i} \left(\beta_{y_i}^T \mathbf{x}_i - \log\left(\sum_{j=1}^{K} e^{\beta_j^T \mathbf{x}_i} \right) \right) \tag{5.6}$$

In order to find the parameters that maximize the log-likelihood on the training set the Limited-memory BFGS (LBFGS) optimization algorithm was used with an iteration parameter of a thousand steps. LBFGS belongs to the Quasi-Newton methods and works well on large-scale optimization problems with a high number of variables. The number of variables that were trained on was 6 logits + 132 features = 138 variables. This method maintains a smaller memory of updates in the algorithm making it memory efficient.

5.3. Results

5.3.1. Performance on ROC curve

Figure 5.3 shows the ROC curves of Sparcnet 1, MLR and the Boosting Cascade with 40 features. The MLR model shows better AUC on all classes except LPD. On Other, Seizure, LRDA and GRDA the MLR curve is completely above Sparcnet's curve with considerably higher AUC. The Cascade with 40 features has comparable curves as Sparcnet with minor improvement on Other and Seizure and minor decline in AUC for LPD and GPD. In the ROC curves the Multinomial Logistic Regression has a higher AUC than both the Cascade and Sparcnet with the exception of the LPD class.

5.3.2. Performance on PR curve

In the PR curves of figure 5.4 the boosting cascade with 40 features only shows improvements in AUC on the Other and Seizure class. This means that most of the reclassifications from one of the "non Other" classes were done on samples that had Seizure labels. The cascade had a minimal impact on the AUC of "Other" but a larger impact on the Seizure AUC. The MLR model shows a considerable

improvement for ”Other” AUC from 0.82 to 0.91. It has a slightly lower AUC on LPD and GPD while having a slightly better AUC in the PR curve for ”Seizure”, ”LRDA”, and ”GRDA”.

5.3.3. Macro Evaluation Metrics

The macro evaluation metrics in table 5.1, which are calculated after taking the maximum probability for a class, show a small improvement in the accuracy by applying the boosting cascade. This difference in accuracy can be explained by the over sampled ”Other” class and the reclassification to Other. The cascade has the highest score for both the balanced metrics macro F1 and MCC. While the metrics are inconsistent for the ”Other” class the changes in evaluation metrics per class are minor with the exception of the Seizure class, just as in the PR curve. The seizure class has a considerably larger precision and about the same recall resulting in a higher F1 score. The MLR model shows lower accuracy than the two other methods but a higher macro recall. The higher recall is especially larger for the ”GPD”, ”GRDA” and ”LRDA” classes.

5.3.4. Confusion Matrices

Figures 5.5 and 5.6 show the confusion matrices with the percentage of actual labels and the predicted labels from the Boosting Cascade and MLR models. The confusion matrices in figure 5.6 show that for the MLR model, LRDA is confused the least with 12% being misclassified as Other, 2% as Seizure and LPD, and 9% as GRDA. Noticeably LRDA is never confused with GPD which is explainable since the pattern is different and the localization of the pattern is also different even though 1% of GPD’s is being misclassified as LRDA. In the MLR confusion matrix of figure 5.5, Other and GRDA are the second best accurately classified classes with 56% of actual labels being predicted correctly. After these two classes, LPDs are classified accurately in 51%, Seizures in 44%, and GPD in 40% of the cases. GPDs are often confused with LPDs and Seizures are often confused with LPDs. This means that in terms of actual classes and predictions, the MLR recalibrated towards better detection of ”LRDA” and ”GRDA” at the cost of higher false positives for these classes.

5.3.5. Swimmer plots

The swimmer plots of patients ”glad064”, ”glad024”, ”glad132” and ”glad105” can be seen in figures 5.7 and 5.8. These patients were chosen because they showed interesting behavior from the models in the swimmer plots. The swimmer plots from patient number 24 in the test set (glad024) show that the Boosting Cascade and MLR clearly are capable of rejecting the false positive seizures that were caused by artifacts. Sparcnet 1 classifies a large number of samples as the ”Seizure” class in the swimmer plot as can be seen by all the red bars. Both the Boosting Cascade and the MLR reclassify most of the labels as ”Other”. This effect is also shown in patient number 64 from the test set (glad064) for the Boosting Cascade. Sparcnet 1 has some ”Seizure” false positives halfway and near the end of the EEG recording. These labels are reclassified to the ”Other” class as well. MLR reclassifies most of the cases in the middle where there is an increase in power across the frequency bands as the ”LPD” class which is not in line with either the expert or BOW labels. The model does seem to classify more ”GPD” classes before and after this segment which is correct compared to the expert and BOW labels but at the cost of more false positives for the other classes. The swimmer plots of ”glad132” show that when the spectrograms are not exhibiting extreme dB values there is almost no change in the labels compared to Sparcnet 1. MLR shows a slight improvement in the classification of the ”LPD” class near the end of the recording but again at the cost of extra false positives. The same thing for the Boosting cascade happens on ”glad105”, there is not a single change to the Sparcnet 1 classifications. MLR on the other hand increased the detection of the ”GPD” class.

Also notable is that there is an indication that the Bag of Words (BOW) labels are not completely correct for ”glad064.” The swimmer plot in Figure 5.7 shows that on the segment where there is an increase in power in the spectrogram, a number of ”GPD” labels can be seen. Figure 5.9 shows a raw EEG sample from one of these segments that illustrates that these samples experience artifacts and an argument could be made that they belong to the ”Other” class.

5.4. Discussion

The Boosting Cascade shows better overall macro evaluation metrics than Sparcnet 1. It not only has a better accuracy but also a better precision and recall trade-off although the margins of improvement

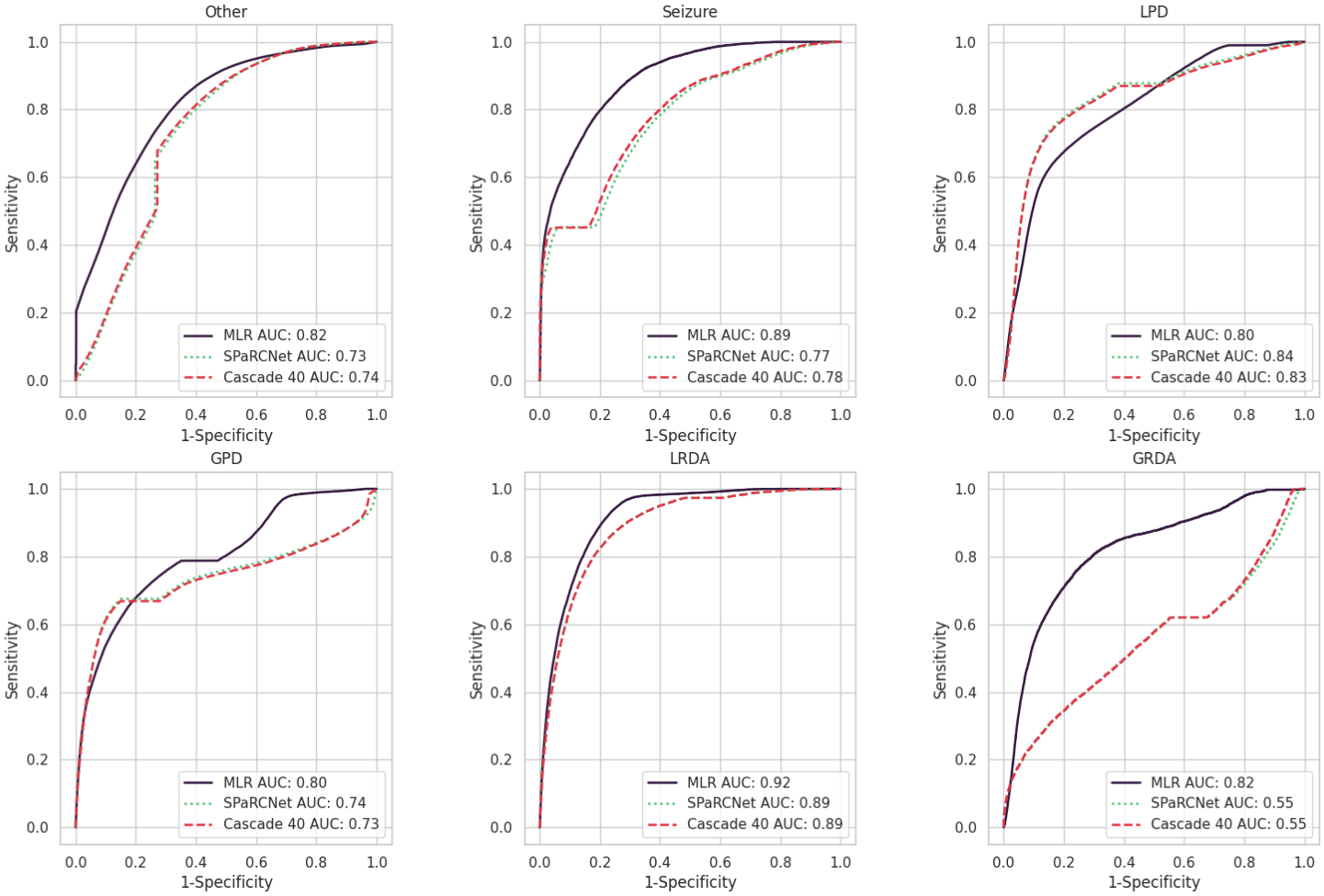


Figure 5.3: ROC curves for feature based models

are not really large. It reduces the False Positive Rate for Seizure, LPD, GPD, and LRDA while slightly increasing the False Positive Rate for the other class. The greatest improvement is on the Seizure class which is a reduction of 34.7%. The method has also shown to be able to reject clear artifacts based on the spectrograms as seen in figure 5.7. This means that, at the cost of increasing the False Positive Rate with 2.4%, the Boosting Cascade is able to at least improve the detection of the other class and reject some of the False Positives. The only false positive reductions that show up in the normalized confusion matrices are for the Seizure and LPD class which shows that the improvements are not that significant on GPD and LRDA. These results show that this method albeit by a small amount is able to reduce the false positive rate of the classifications on noisy continuous EEG recordings at the cost of slightly increasing the false positive rate of the Other class. The swimmer plots indicate that even though there is an improvement, the model seems to reclassify clear artifacts and does not really reduce the false positive rate of the other classes that much.

The Multinomial Logistic regression model used to re-calibrate the logits of Sparcnet and the features from the boosting cascade has the lowest macro precision but the largest recall. This means that in general over all classes the model is more sensitive to detecting the actual positives. The precision and recall showed that MLR is relatively good at predicting the ”Other” and ”Seizure” classes. This is further supported by their low class-specific False Positive Rates. On all the other classes the False Positive Rate is higher than Sparcnet 1 and the Boosting Cascade, this is in line with the lower macro precision and higher macro recall. If specifically a low False Positive Rate for seizures is required then this is a good postprocessing method. When the model is forced to make predictions based on the highest output probability there is a clear increase in the recall for the GPD, LRDA, and GRDA which means it captures more of the actual cases from these classes. Based on the ROC curves MLR is bet-

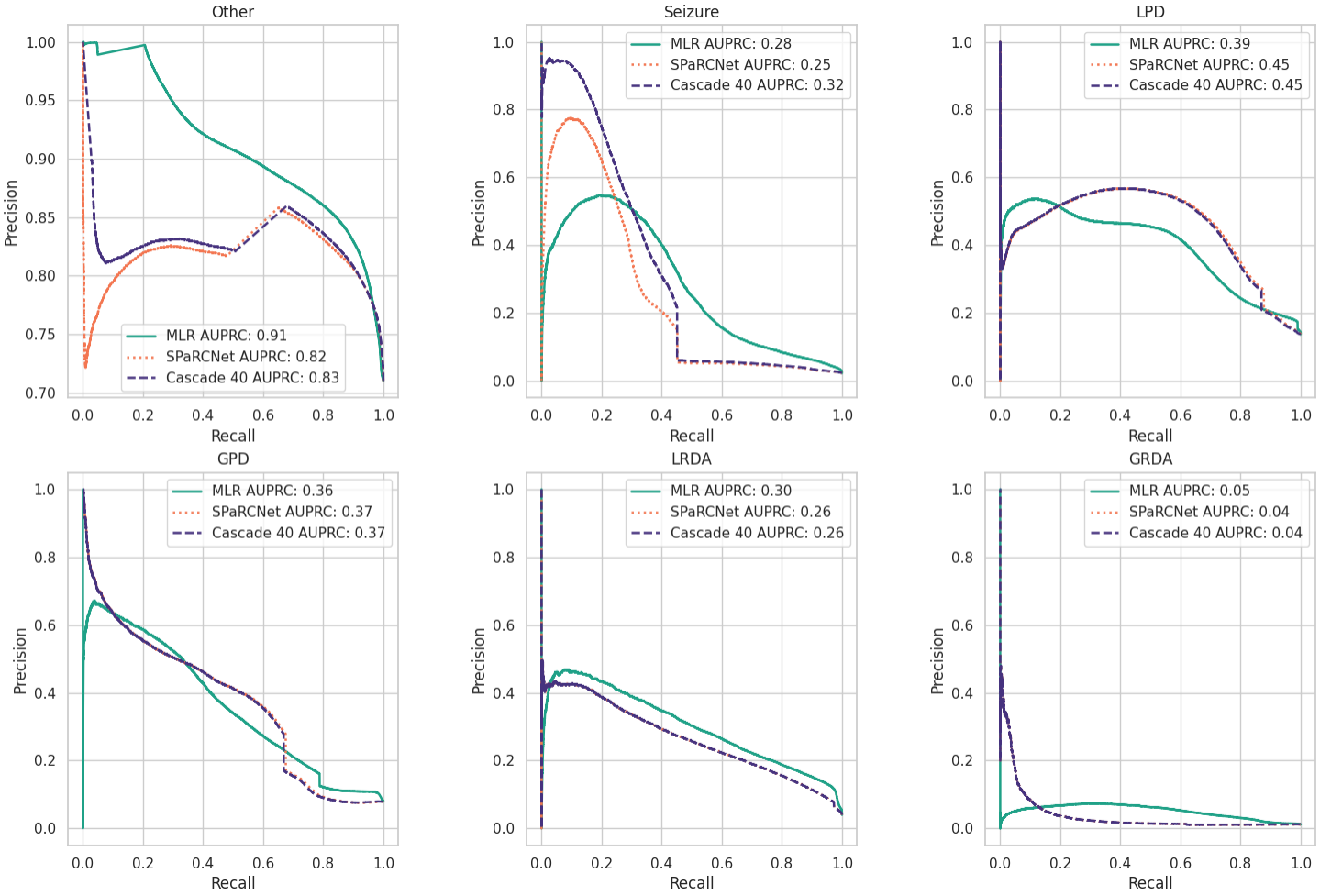


Figure 5.4: PR curves for feature based models

ter at distinguishing between if the specific class is present or not over all thresholds with the exception of the LPD class.

Metric	Sparcnet 1	MLR	Cascade 40
Macro Metrics			
Accuracy	0.6369	0.5432	**0.6561**
Precision Macro	0.3884	0.3866	**0.3998**
Recall Macro	0.4526	**0.5398**	0.4558
F1 Macro	0.3848	0.3921	**0.3981**
MCC	0.3425	0.3306	**0.3572**
Other Class			
Precision	0.8410	**0.8968**	0.8431
Recall	0.7112	0.5559	**0.7397**
F1	0.7706	0.6863	**0.7880**
FPR	0.3295	**0.1567**	0.3373
Seizure Class			
Precision	0.1587	**0.3288**	0.2239
Recall	**0.4831**	0.4444	0.4823
F1	0.2389	**0.3779**	0.3059
FPR	0.0598	**0.0212**	0.0390
LPD Class			
Precision	**0.5355**	0.4078	0.5353
Recall	**0.5711**	0.5100	0.5642
F1	**0.5527**	0.4532	0.5494
FPR	0.0788	0.1178	**0.0779**
GPD Class			
Precision	0.5036	0.4221	**0.5046**
Recall	0.2749	**0.4014**	0.2729
F1	0.3556	**0.4115**	0.3542
FPR	0.0232	0.0470	**0.0229**
LRDA Class			
Precision	0.2565	0.2194	**0.2567**
Recall	0.4656	**0.7642**	0.4656
F1	0.3307	**0.3410**	0.3309
FPR	0.0558	0.1123	**0.0557**
GRDA Class			
Precision	0.0349	**0.0446**	0.0350
Recall	0.2100	**0.5631**	0.2100
F1	0.0599	**0.0826**	0.0599
FPR	**0.0670**	0.1395	**0.0670**

Table 5.1: Comparison of Sparcnet 1, MLR, and Cascade 40 Models

5.5. Limitations

Recalibrating the output of Sparcnet 1 with the Boosting Cascade and MLR can improve certain performance metrics but there are limitations. These methods are highly dependent on the quality of the output from Sparcnet 1. Transfer learning does not have this problem because the Sparcnet 1 model can provide a good starting point even when the initial model is not optimal. This means that the problems of the initial model will be propagated through the recalibrated outputs from the Boosting Cascade and the MLR. Furthermore, logistic regression and the decision thresholds from the boosting cascade might not be able to capture the complex non-linear patterns that the DenseNet Convolutional Neural Network was designed to capture. Logistic regression is a generalized linear model and captures non-linear patterns less well than deep learning models. This is because logistic regression constructs linear boundaries to separate the data. Since the electrical activity measured in EEG exhibits complex behavior with nonlinear dynamic properties more complex models might be better suited. Finally, Sparcnet 1 learned rich hierarchical features that were used to capture nuanced patterns in the data. These features can become diluted or overlooked by simpler methods like logistic regression or a boosting cascade.

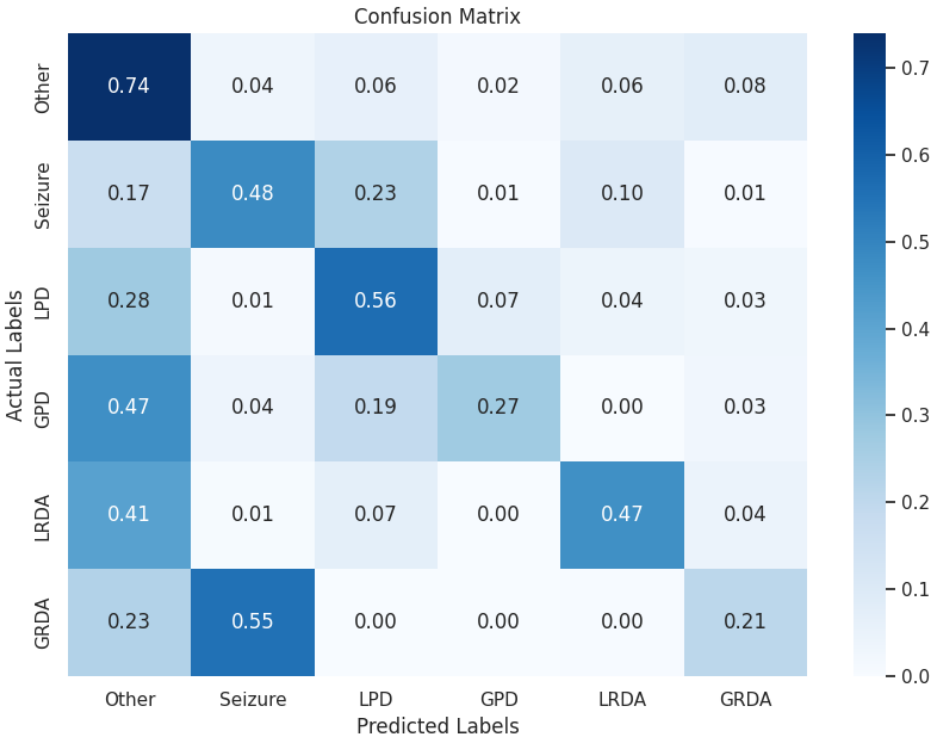


Figure 5.5: Confusion matrix Cascade 40

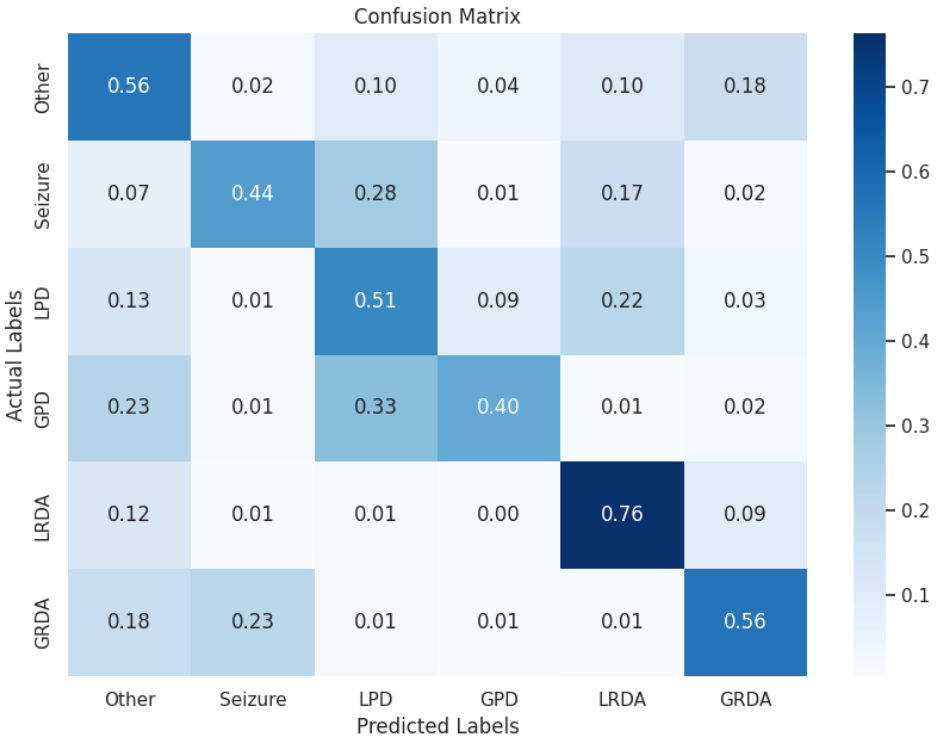


Figure 5.6: Confusion matrix Multinomial Logistic Regression

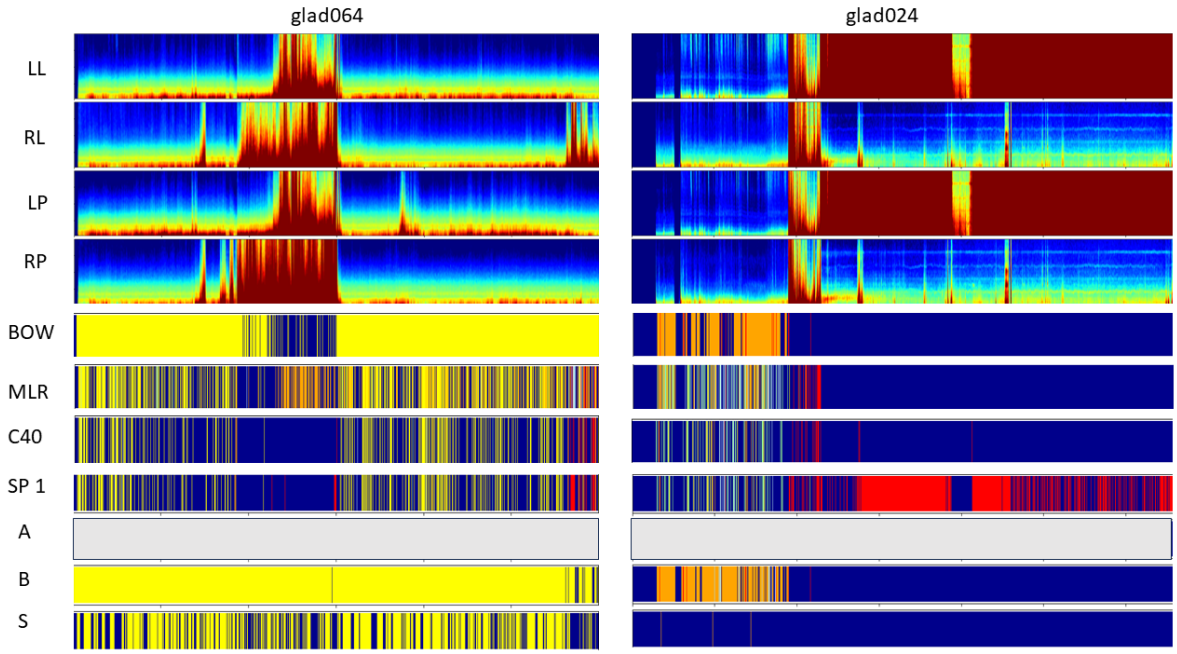


Figure 5.7: glad004 and glad017 postprocessing swimmer plots and spectrograms

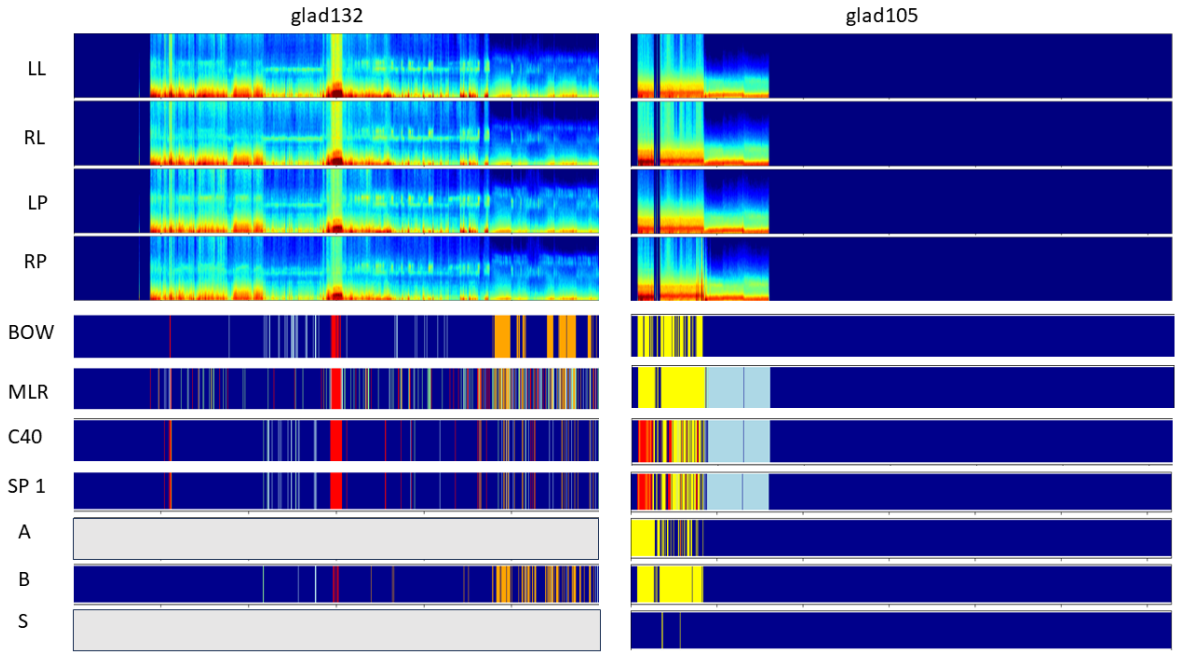


Figure 5.8: glad132 and glad105 postprocessing swimmer plots and spectrograms

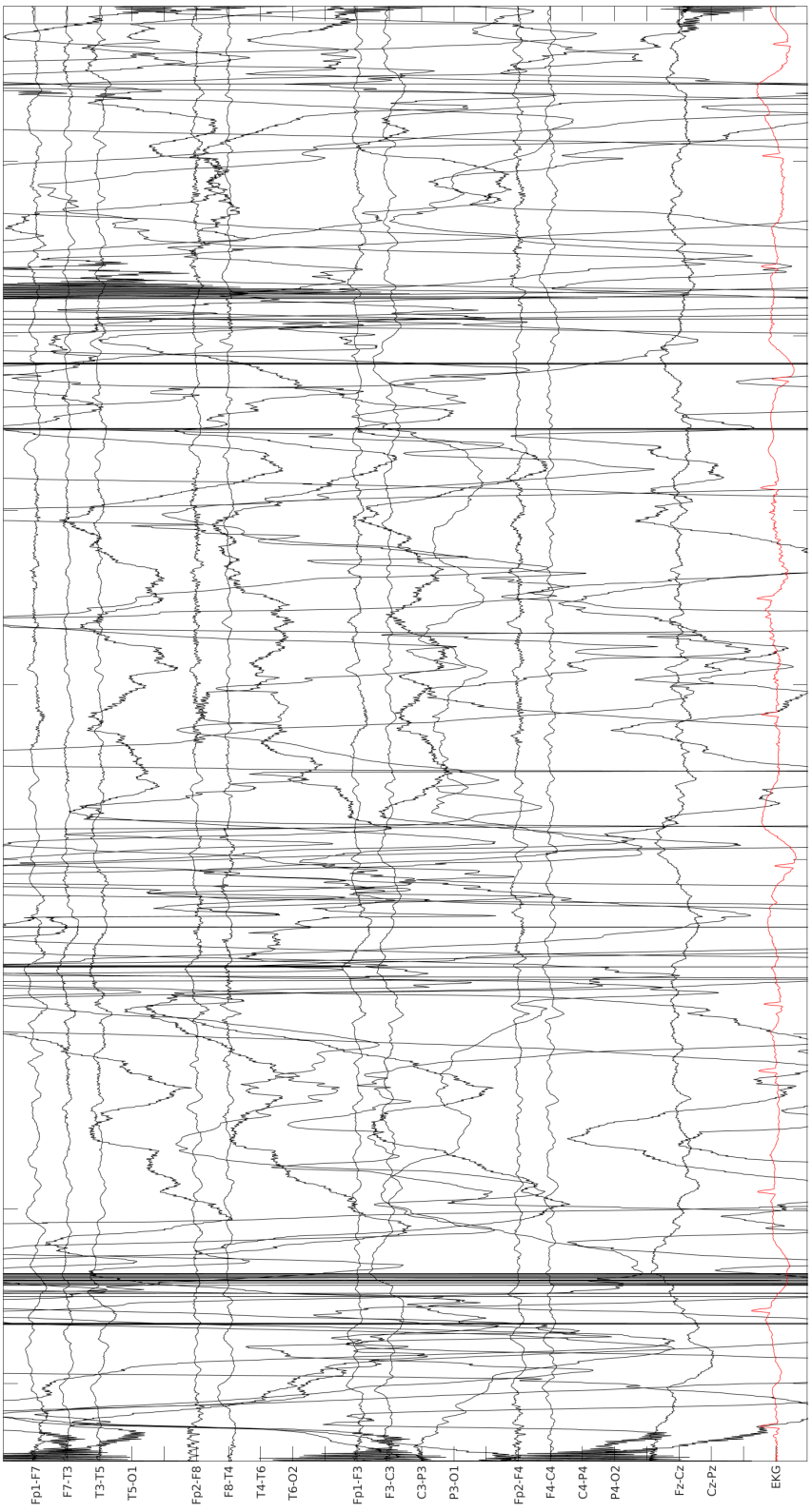


Figure 5.9: Raw EEG sample from glad064

6

General Discussion

6.1. Summary of Results

The frequency content of the EEG signal can improve labels in some cases through Bag of Words and Change Point Detection. The parameters have to be tuned per patient by an expert and in some cases, the labels from the pipeline have to be overwritten by expert labels or a combination of expert labels. The method worked on patients with segments that showed a clear increase in power over the frequency ranges in different brain regions. This method holds the potential to improve labels over a short period of time and the GUI can serve as a tool to quickly validate label quality.

The transfer learning experiments failed to outperform Sparcnet 1. Re-training more layers resulted in worse ROC curves, PR curves, and evaluation metrics (figures 4.3, 4.4 and table 4.4). Training the entire model, denseblock 7 and not initializing Sparcnet's weight resulted in training loss curves that did not show any learning. Once trained these models each omitted one of the classes entirely from their classifications. Training the classifier from the DenseNet model with the weights from Sparcnet 1 gave the best results. The best model was the model with a batch size of 256, as this change in batch size significantly increased the macro evaluation metrics compared to a model with a batch size of 128. This model has a strong bias towards the ”LRDA” class as seen in the confusion matrix (figure 4.6). The training hyperparameters were shown to have an effect on the training loss curves and on evaluation metrics. Any changes to the learning rate or data shuffling from Sparcnet 1 resulted in a model not classifying one of the classes and a great reduction in performance on the evaluation metrics, ROC curve, and PR curve.

The boosting cascade with 40 features showed an increase in performance on macro evaluation metrics compared to Sparcnet 1. The model increased the False Positive rate for the ”Other” class but reduced the false positive rate for the rest of the classes. The improvements were extremely small but the swimmer plots did show the ability to recognize clear artifacts and reclassify the ”Seizure” class as ”Other. Using MLR as a postprocessing method with the logits from Sparcnet 1 and the features from the boosting cascade gave better ROC curves than both Sparcnet 1 and the Boosting Cascade with the exception of the ”LPD” class. MLR resulted in having the lowest macro precision but the largest macro recall. The model showed an increase in false positives for all the classes except the ”Other” and ”Seizure” classes, reducing their false positive rate significantly.

6.2. Summary of Limitations

The Bag of Words method has two limitations. First, the method has to be tuned by an expert and can not be generalized across all patients. Since the expert has to both set the parameters and the decision rules on which labels to keep, the method is susceptible to the bias of the expert. In this case, the pipeline was tuned by one expert and there could be a preference towards their own labels. Second, the Bag of Words method only clusters on power spectral density. This ensures capturing spectral information but temporal, spatial, and morphological information from the EEG signal is overlooked.

The transfer learning experiments are highly dependent on the label quality and on the choices in training hyperparameters. The swimmer plot from ”glad017” (figure 5.7) indicated that the limitations of the Bag of Words method could have an influence on the results from the transfer learning experiments.

This could mean that there are other incorrectly labeled segments like in ”glad017” in the training and test data. These incorrectly labeled segments could add too much noise for the models to learn the correct patterns and the evaluation metrics could be based on wrong labels. Training a DenseNet Convolutional Neural Network on this much data is computationally intensive. Because of a lack of resources and time not all parameters were chosen to be optimal. The implementation of an adaptive learning rate, k-fold cross-validation, and a grid search across parameters were not conducted.

Multinomial Logistic Regression and the Boosting Cascade rely heavily on the quality of the outputs from Sparcnet 1. The problems of the initial model will be passed on to the calibrated outputs. Unlike in transfer learning where the weights from Sparcnet 1 function as a starting point and initialization even when the model is not optimal. The threshold from the Boosting Cascade and the logistic regression model might not be able to capture the complex nonlinear patterns that are present in the EEG data and the hierarchical features from Sparcnet 1 could be lost.

6.3. Context to related work

This research is the first work that tries retraining Sparcnet 1 for better performance on noisy long continuous EEG recordings, the only model capable of capturing Seizure and IIIC patterns in EEG recordings of critically ill patients in the ICU. It tries to overcome the limited data availability by using the Bag of Words model to create a new large dataset and use transfer learning to initialize the model with the information already learned by Sparcnet 1. By retraining Sparcnet 1 this research continues the work of Jin Jing et al. [30]. Through the Bag of Words and transfer learning methods, steps were taken to create a dataset with high-quality labels and a model that can detect Seizure and IIIC patterns in longer noisy continuous EEG recordings from a larger set of patients. Furthermore, two feature-based postprocessing methods were applied to the output of Sparcnet 1 in continuation of the work by Dirks et al. [15]. Even though this work did not significantly improve the performance of Sparcnet 1 it succeeded in identifying major problems and limitations and can be used as a stepping stone for future work.

6.4. Future work

Label improvement

The main route forward is improving the quality of the labels. There were enough indications that the labels were not where they needed to be for training deep learning models. The first step would be to have the experts do another round of labeling and let them validate their labels with the Bag of Words tool. These new labels can serve again as the input for the Bag of Words pipeline which becomes better with higher quality labels. More research can also be done on using the features from the Boosting Cascade for clustering in Bag of Words instead of power spectral density values. Using the features in the pipeline would solve the limitation of the method and also cover temporal and morphological information of the EEG signal.

Hyperparameters

Improvements can be made by implementing the proposed solutions to deal with data generalization, learning rate, and hyperparameters. Adding the adaptive learning rate, k-fold cross-validation, and a gridsearch could improve the performance of the models. Furthermore, the early stopping logic greatly influences model training, and the shuffle parameter, although computationally intensive, is a robust way to ensure that the model does not learn the order of the data. An interesting experiment would be to remove the early stopping logic, add shuffling, and let the model train for 500 epochs while monitoring the training and validation loss. Experimenting with and implementing these strategies is a way to exclude hyperparameters as the cause of the bad performance of the deep learning models. Other more advanced hyperparameter tuning solutions include coarse-to-fine random searches and Bayesian hyperparameter optimization. The idea of coarse-to-fine search is to perform a random search on the initial hyperparameter space, find a promising area, and perform a grid search on that area. At a high level, Bayesian search starts with a prior estimate of parameter distributions, maintains a probabilistic model of the relationship between hyperparameter values and model performance, and then alternates between training and maximizing the estimate and updating the probabilistic model.

High Performance Computing cluster

In order to make these suggested improvements, that add a lot of computational operations, there needs to be enough computing power. Training these deep learning neural networks is best done in a cycle of tuning, improving labels, and training. In this work, the cycle has been completed once but it is highly recommended to speed up this process by training on a High Performance Computing (HPC) cluster. The reason for this is because of the amount of data in the continuous long EEG recordings and the workload of training a DenseNet CNN. HPC clusters consist of high speed computer servers that are networked together and have a centralized scheduler that can efficiently manage parallel computing workload.

Class imbalance

The data is highly imbalanced and the solution currently is to use a weighted loss function which is an algorithm level approach to the problem but there are also data level and hybrid solutions. More research could be done by exploring data level solutions with a different loss function. This would entail using over sampling, under sampling, or SMOTE while training with another loss function like cross entropy loss.

Include more patterns

A large limitation for all the methods is that the current ”Other” class encompasses everything that is not a seizure or an IIIC pattern. This class is broad and includes for example patterns that are seen as normal behavior and different artifacts caused by muscles, electrical interference, noise, and movements. Feeding the deep-learning model examples with these two completely different patterns with the same label could confuse the model. An important recommendation would be to split up the ”Other” class. The data showed to have many artifacts and the models should be able to distinguish a normal EEG and an artifact.

References

[1] Eugenio Abela et al. “Slower alpha rhythm associates with poorer seizure control in epilepsy”. In: *Annals of Clinical and Translational Neurology* 6.2 (Dec. 2018), pp. 333–343. ISSN: 2328-9503. DOI: 10.1002/acn3.710. URL: http://dx.doi.org/10.1002/acn3.710.

[2] Dimitrios Adamis, Sunita Sahu, and Adrian Treloar. “The utility of EEG in dementia: a clinical perspective”. In: *International Journal of Geriatric Psychiatry* 20.11 (2005), pp. 1038–1045. ISSN: 1099-1166. DOI: 10.1002/gps.1393. URL: http://dx.doi.org/10.1002/gps.1393.

[3] Vincent Alvarez and Andrea O. Rossetti. “Clinical Use of EEG in the ICU: Technical Setting”. In: *Journal of Clinical Neurophysiology* 32.6 (Dec. 2015), pp. 481–485. ISSN: 0736-0258. DOI: 10.1097/wnp.0000000000000194. URL: http://dx.doi.org/10.1097/WNP.0000000000000194.

[4] Ushtar Amin and Selim R. Benbadis. “The Role of EEG in the Erroneous Diagnosis of Epilepsy”. In: *Journal of Clinical Neurophysiology* 36.4 (July 2019), pp. 294–297. ISSN: 0736-0258. DOI: 10.1097/wnp.0000000000000572. URL: http://dx.doi.org/10.1097/WNP.0000000000000572.

[5] Elarbi Badidi. “Edge AI for Early Detection of Chronic Diseases and the Spread of Infectious Diseases: Opportunities, Challenges, and Future Directions”. In: *Future Internet* 15.11 (Nov. 2023), p. 370. ISSN: 1999-5903. DOI: 10.3390/fi15110370. URL: http://dx.doi.org/10.3390/fi15110370.

[6] Erol Başar et al. “Brain’s alpha, beta, gamma, delta, and theta oscillations in neuropsychiatric diseases”. In: *Supplements to Clinical Neurophysiology*. Elsevier, 2013, pp. 19–54. ISBN: 9780702053078. DOI: 10.1016/b978-0-7020-5307-8.00002-8. URL: http://dx.doi.org/10.1016/B978-0-7020-5307-8.00002-8.

[7] Ettore Beghi, Giorgia Giussani, and Nichols. “Global, regional, and national burden of epilepsy, 1990–2016: a systematic analysis for the Global Burden of Disease Study 2016”. In: *The Lancet Neurology* 18.4 (Apr. 2019), pp. 357–375. ISSN: 1474-4422. DOI: 10.1016/s1474-4422(18)30454-x. URL: http://dx.doi.org/10.1016/S1474-4422(18)30454-X.

[8] Ramina Behzad and Aida Behzad. “The Role of EEG in the Diagnosis and Management of Patients with Sleep Disorders”. In: *Journal of Behavioral and Brain Science* 11.10 (2021), pp. 257–266. ISSN: 2160-5874. DOI: 10.4236/jbbs.2021.1110021. URL: http://dx.doi.org/10.4236/jbbs.2021.1110021.

[9] J. Claassen et al. “Detection of electrographic seizures with continuous EEG monitoring in critically ill patients”. In: *Neurology* 62.10 (May 2004), pp. 1743–1748. ISSN: 1526-632X. DOI: 10.1212/01.wnl.0000125184.88797.62. URL: http://dx.doi.org/10.1212/01.wnl.0000125184.88797.62.

[10] Jan Claassen and Paul Vespa. “Electrophysiologic Monitoring in Acute Brain Injury”. In: *Neurocritical Care* 21.S2 (Sept. 2014), pp. 129–147. ISSN: 1556-0961. DOI: 10.1007/s12028-014-0022-8. URL: http://dx.doi.org/10.1007/s12028-014-0022-8.

[11] Jan Claassen et al. “Prognostic Significance of Continuous EEG Monitoring in Patients With Poor-Grade Subarachnoid Hemorrhage”. In: *Neurocritical Care* 4.2 (2006), pp. 103–112. ISSN: 1541-6933. DOI: 10.1385/ncc:4:2:103. URL: http://dx.doi.org/10.1385/ncc:4:2:103.

[12] Justine Cormier, Carolina Maciel, and Emily Gilmore. “Ictal-Interictal Continuum: When to Worry About the Continuous Electroencephalography Pattern”. In: *Seminars in Respiratory and Critical Care Medicine* 38.06 (Dec. 2017), pp. 793–806. ISSN: 1098-9048. DOI: 10.1055/s-0037-1607987. URL: http://dx.doi.org/10.1055/s-0037-1607987.

[13] Leonid Datta. “A Survey on Activation Functions and their relation with Xavier and He Normal Initialization”. In: (2020).

[14] Prashant Deshmukh et al. “Epileptic seizure detection using discrete wavelet transform based support vector machine”. In: *2017 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, Apr. 2017. DOI: 10.1109/iccsp.2017.8286736. URL: http://dx.doi.org/10.1109/ICCSP.2017.8286736.

[15] E.H.M. Dirks et al. *Automated Identification of Ictal-Interictal Injury Continuum Patterns in Continuous EEG Recordings*. Massachusetts General Hospital/Harvard Medical School Department of Neurology, Boston, United States of America; Massachusetts General Hospital Clinical Data Animation Center (CDAC), Boston, United States of America; University of Twente Department of Clinical Neurophysiology, Enschede, The Netherlands; Medical Spectrum Twente Departments of Neurology and Clinical Neurophysiology, Enschede, The Netherlands. Nov. 2022.

[16] Brandon Foreman et al. “Generalized periodic discharges in the critically ill: A case-control study of 200 patients”. In: *Neurology* 79.19 (Nov. 2012), pp. 1951–1960. ISSN: 1526-632X. DOI: 10.1212/wnl.0b013e3182735cd7. URL: http://dx.doi.org/10.1212/wnl.0b013e3182735cd7.

[17] Irene García–Morales et al. “Periodic Lateralized Epileptiform Discharges: Etiology, Clinical Aspects, Seizures, and Evolution in 130 Patients”. In: *Journal of Clinical Neurophysiology* 19.2 (Mar. 2002), pp. 172–177. ISSN: 0736-0258. DOI: 10.1097/00004691-200203000-00009. URL: http://dx.doi.org/10.1097/00004691-200203000-00009.

[18] Nicolas Gaspard et al. “Similarity of Lateralized Rhythmic Delta Activity to Periodic Lateralized Epileptiform Discharges in Critically Ill Patients”. In: *JAMA Neurology* (Aug. 2013). ISSN: 2168-6149. DOI: 10.1001/jamaneurol.2013.3475. URL: http://dx.doi.org/10.1001/jamaneurol.2013.3475.

[19] Wendong Ge et al. “Deep active learning for interictal ictal injury continuum EEG patterns”. In: *Journal of Neuroscience Methods* 351 (Mar. 2021), p. 108966. ISSN: 0165-0270. DOI: 10.1016/j.jneumeth.2020.108966. URL: http://dx.doi.org/10.1016/j.jneumeth.2020.108966.

[20] Zakary Georgis-Yap, Milos R. Popovic, and Shehroz S. Khan. “Supervised and Unsupervised Deep Learning Approaches for EEG Seizure Prediction”. In: (2023). DOI: 10.48550/ARXIV.2304.14922. URL: https://arxiv.org/abs/2304.14922.

[21] Christoph Golz et al. “Preparing students to deal with the consequences of the workforce shortage among health professionals: a qualitative approach”. In: *BMC Medical Education* 22.1 (Nov. 2022). ISSN: 1472-6920. DOI: 10.1186/s12909-022-03819-4. URL: http://dx.doi.org/10.1186/s12909-022-03819-4.

[22] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[23] Rabel Guharoy, Nanda Dulal Jana, and Suparna Biswas. “An Efficient Epileptic Seizure Detection Technique using Discrete Wavelet Transform and Machine Learning Classifiers”. In: *Journal of Physics: Conference Series* 2286.1 (July 2022), p. 012013. ISSN: 1742-6596. DOI: 10.1088/1742-6596/2286/1/012013. URL: http://dx.doi.org/10.1088/1742-6596/2286/1/012013.

[24] Lawrence J. Hirsch et al. “American Clinical Neurophysiology Society’s Standardized Critical Care EEG Terminology: 2021 Version”. In: *Journal of Clinical Neurophysiology* 38.1 (Jan. 2021), pp. 1–29. ISSN: 0736-0258. DOI: 10.1097/wnp.0000000000000806. URL: http://dx.doi.org/10.1097/wnp.0000000000000806.

[25] Gao Huang et al. *Densely Connected Convolutional Networks*. 2016. DOI: 10.48550/ARXIV.1608.06993. URL: https://arxiv.org/abs/1608.06993.

[26] S. Indolia et al. “Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach”. In: *Procedia Computer Science* (2018).

[27] Corentin Jacques et al. “Low and high frequency intracranial neural signals match in the human associative cortex”. In: *eLife* 11 (Sept. 2022). ISSN: 2050-084X. DOI: 10.7554/elife.76544. URL: http://dx.doi.org/10.7554/eLife.76544.

[28] Suparerk Janjarasjitt. “Epileptic seizure classifications of single-channel scalp EEG data using wavelet-based features and SVM”. In: *Medical amp; Biological Engineering amp; Computing* 55.10 (Feb. 2017), pp. 1743–1761. ISSN: 1741-0444. DOI: 10.1007/s11517-017-1613-2. URL: http://dx.doi.org/10.1007/s11517-017-1613-2.

[29] Yating Jiang, Lingling Yang, and Yao Lu. “An Epileptic Seizure Prediction Model based on a Simulation Block and a Pretrained ResNet”. In: *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. IEEE, Oct. 2020. DOI: 10.1109/cisp-bmei51763.2020.9263591. URL: http://dx.doi.org/10.1109/CISP-BMEI51763.2020.9263591.

[30] Jin Jing et al. “Development of Expert-Level Classification of Seizures and Rhythmic and Periodic Patterns During EEG Interpretation”. In: *Neurology* 100.17 (Apr. 2023). ISSN: 1526-632X. DOI: 10.1212/wnl.0000000000207127. URL: http://dx.doi.org/10.1212/wnl.0000000000207127.

[31] Jin Jing et al. “Rapid Annotation of Seizures and Interictal-ictal Continuum EEG Patterns”. In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, July 2018. DOI: 10.1109/embc.2018.8513059. URL: http://dx.doi.org/10.1109/embc.2018.8513059.

[32] Jin Jing et al. “Rapid annotation of seizures and interictal-ictal-injury continuum EEG patterns”. In: *Journal of Neuroscience Methods* 347 (Jan. 2021), p. 108956. ISSN: 0165-0270. DOI: 10.1016/j.jneumeth.2020.108956. URL: http://dx.doi.org/10.1016/j.jneumeth.2020.108956.

[33] Emily L. Johnson and Peter W. Kaplan. “Population of the ictal-interictal zone: The significance of periodic and rhythmic activity”. In: *Clinical Neurophysiology Practice* 2 (2017), pp. 107–118. ISSN: 2467-981X. DOI: 10.1016/j.cnp.2017.05.001. URL: http://dx.doi.org/10.1016/j.cnp.2017.05.001.

[34] Colin B. Josephson, Susan Rahey, and R. Mark Sadler. “Neurocardiogenic Syncope: Frequency and Consequences of its Misdiagnosis as Epilepsy”. In: *Canadian Journal of Neurological Sciences / Journal Canadien des Sciences Neurologiques* 34.2 (May 2007), pp. 221–224. ISSN: 2057-0155. DOI: 10.1017/s0317167100006089. URL: http://dx.doi.org/10.1017/S0317167100006089.

[35] Joon Y. Kang and Gregory L. Krauss. “Normal Variants Are Commonly Overread as Interictal Epileptiform Abnormalities”. In: *Journal of Clinical Neurophysiology* 36.4 (July 2019), pp. 257–263. ISSN: 0736-0258. DOI: 10.1097/wnp.0000000000000613. URL: http://dx.doi.org/10.1097/WNP.0000000000000613.

[36] Alexander Ya. Kaplan et al. “Nonstationary nature of the brain activity as revealed by EEG/MEG: Methodological, practical and conceptual challenges”. In: *Signal Processing* 85.11 (Nov. 2005), pp. 2190–2212. ISSN: 0165-1684. DOI: 10.1016/j.sigpro.2005.07.010. URL: http://dx.doi.org/10.1016/j.sigpro.2005.07.010.

[37] Santhosh Karnik, Justin Romberg, and Mark A. Davenport. “Thomson’s Multitaper Method Revisited”. In: *IEEE Transactions on Information Theory* 68.7 (July 2022), pp. 4864–4891. ISSN: 1557-9654. DOI: 10.1109/tit.2022.3151415. URL: http://dx.doi.org/10.1109/TIT.2022.3151415.

[38] Haidar Khan et al. “Focal Onset Seizure Prediction Using Convolutional Networks”. In: *IEEE Transactions on Biomedical Engineering* 65.9 (Sept. 2018), pp. 2109–2118. ISSN: 1558-2531. DOI: 10.1109/tbme.2017.2785401. URL: http://dx.doi.org/10.1109/TBME.2017.2785401.

[39] R. Killick, P. Fearnhead, and I. A. Eckley. “Optimal detection of changepoints with a linear computational cost”. In: (2011). DOI: 10.48550/ARXIV.1101.1438. URL: https://arxiv.org/abs/1101.1438.

[40] Hee E. Kim et al. “Transfer learning for medical image classification: a literature review”. In: *BMC Medical Imaging* 22.1 (Apr. 2022). ISSN: 1471-2342. DOI: 10.1186/s12880-022-00793-7. URL: http://dx.doi.org/10.1186/s12880-022-00793-7.

[41] Hyo-Eun Kim et al. “Changes in cancer detection and false-positive recall in mammography using artificial intelligence: a retrospective, multireader study”. In: *The Lancet Digital Health* 2.3 (Mar. 2020), e138–e148. ISSN: 2589-7500. DOI: 10.1016/s2589-7500(20)30003-0. URL: http://dx.doi.org/10.1016/S2589-7500(20)30003-0.

[42] Fumio Kubota and Naoki Ohnishi. “Study on FIRDA and 3 Hz Rhythmic Slow Wave Bursts Occurring in the Frontal Area of Epileptic Patients”. In: *Clinical Electroencephalography* 28.2 (Apr. 1997), pp. 112–116. ISSN: 0009-9155. DOI: 10.1177/155005949702800209. URL: http://dx.doi.org/10.1177/155005949702800209.

[43] Pedro Kurtz et al. “Continuous electroencephalography in a surgical intensive care unit”. In: *Intensive Care Medicine* 40.2 (Nov. 2013), pp. 228–234. ISSN: 1432-1238. DOI: 10.1007/s00134-013-3149-8. URL: http://dx.doi.org/10.1007/s00134-013-3149-8.

[44] Salim Lahmiri and Amir Shmuel. “Accurate Classification of Seizure and Seizure-Free Intervals of Intracranial EEG Signals From Epileptic Patients”. In: *IEEE Transactions on Instrumentation and Measurement* 68.3 (Mar. 2019), pp. 791–796. ISSN: 1557-9662. DOI: 10.1109/tim.2018.2855518. URL: http://dx.doi.org/10.1109/TIM.2018.2855518.

[45] Dohyun Lee et al. “A ResNet-LSTM hybrid model for predicting epileptic seizures using a pre-trained model with supervised contrastive learning”. In: *Scientific Reports* 14.1 (Jan. 2024). ISSN: 2045-2322. DOI: 10.1038/s41598-023-43328-y. URL: http://dx.doi.org/10.1038/s41598-023-43328-y.

[46] Mingyang Li, Wanzhong Chen, and Tao Zhang. “Automatic epilepsy detection using wavelet-based nonlinear analysis and optimized SVM”. In: *Biocybernetics and Biomedical Engineering* 36.4 (2016), pp. 708–718. ISSN: 0208-5216. DOI: 10.1016/j.bbe.2016.07.004. URL: http://dx.doi.org/10.1016/j.bbe.2016.07.004.

[47] Guoyang Liu, Lan Tian, and Weidong Zhou. “Patient-Independent Seizure Detection Based on Channel-Perturbation Convolutional Neural Network and Bidirectional Long Short-Term Memory”. In: *International Journal of Neural Systems* 32.06 (Nov. 2021). ISSN: 1793-6462. DOI: 10.1142/s0129065721500519. URL: http://dx.doi.org/10.1142/S0129065721500519.

[48] Robert Lund et al. “Changepoint Detection in Periodic and Autocorrelated Time Series”. In: *Journal of Climate* 20.20 (Oct. 2007), pp. 5178–5190. ISSN: 0894-8755. DOI: 10.1175/jcli4291.1. URL: http://dx.doi.org/10.1175/JCLI4291.1.

[49] Tsubasa Mawatari et al. “The effect of deep convolutional neural networks on radiologists’ performance in the detection of hip fractures on digital pelvic radiographs”. In: *European Journal of Radiology* 130 (Sept. 2020), p. 109188. ISSN: 0720-048X. DOI: 10.1016/j.ejrad.2020.109188. URL: http://dx.doi.org/10.1016/j.ejrad.2020.109188.

[50] Robert P. McInnis et al. “Epilepsy diagnosis using a clinical decision tool and artificially intelligent electroencephalography”. In: *Epilepsy amp; Behavior* 141 (Apr. 2023), p. 109135. ISSN: 1525-5050. DOI: 10.1016/j.yebeh.2023.109135. URL: http://dx.doi.org/10.1016/j.yebeh.2023.109135.

[51] Johns Hopkins Medicine. *Neurological Conditions*. 2023. URL: https://www.hopkinsmedicine.org/health/conditions-and-diseases/neurological-disorders.

[52] Ana-Claire Meyer et al. “Global disparities in the epilepsy treatment gap: a systematic review”. In: *Bulletin of the World Health Organization* 88.4 (Sept. 2009), pp. 260–266. ISSN: 0042-9686. DOI: 10.2471/blt.09.064147. URL: http://dx.doi.org/10.2471/BLT.09.064147.

[53] Yoav Mintz and Ronit Brodie. “Introduction to artificial intelligence in medicine”. In: *Minimally Invasive Therapy amp; Allied Technologies* 28.2 (Feb. 2019), pp. 73–81. ISSN: 1365-2931. DOI: 10.1080/13645706.2019.1575882. URL: http://dx.doi.org/10.1080/13645706.2019.1575882.

[54] Milad Mirbabaie, Stefan Stieglitz, and Nicholas R. J. Frick. “Artificial intelligence in disease diagnostics: A critical review and classification on the current state of research guiding future direction”. In: *Health and Technology* 11.4 (May 2021), pp. 693–731. ISSN: 2190-7196. DOI: 10.1007/s12553-021-00555-5. URL: http://dx.doi.org/10.1007/s12553-021-00555-5.

[55] Lukas Mosser. “Stochastic Reconstruction of an Oolitic Limestone by Generative Adversarial Network”. In: *Scientific Figure on ResearchGate* (2021).

[56] Nina Moutonnet et al. *Clinical translation of machine learning algorithms for seizure detection in scalp electroencephalography: a systematic review*. 2024. DOI: 10.48550/ARXIV.2404.15332. URL: https://arxiv.org/abs/2404.15332.

[57] Marcus C. Ng, Jin Jing, and M. Brandon Westover. “A Primer on EEG Spectrograms”. In: *Journal of Clinical Neurophysiology* 39.3 (Sept. 2021), pp. 177–183. ISSN: 0736-0258. DOI: 10.1097/wnp.0000000000000736. URL: http://dx.doi.org/10.1097/WNP.0000000000000736.

[58] Soheyl Noachtar and Jan Rémi. “The role of EEG in epilepsy: A critical review”. In: *Epilepsy amp; Behavior* 15.1 (May 2009), pp. 22–33. ISSN: 1525-5050. DOI: 10.1016/j.yebeh.2009.02.035. URL: http://dx.doi.org/10.1016/j.yebeh.2009.02.035.

[59] Isaac Kofi Nti, Owusu Nyarko-Boateng, and Justice Aning. “Performance of Machine Learning Algorithms with Different K Values in K-fold CrossValidation”. In: *International Journal of Information Technology and Computer Science* 13.6 (Dec. 2021), pp. 61–71. ISSN: 2074-9015. DOI: 10.5815/ijitcs.2021.06.05. URL: http://dx.doi.org/10.5815/ijitcs.2021.06.05.

[60] Maria (Meritxell) Oto. “The misdiagnosis of epilepsy: Appraising risks and managing uncertainty”. In: *Seizure* 44 (Jan. 2017), pp. 143–146. ISSN: 1059-1311. DOI: 10.1016/j.seizure.2016.11.029. URL: http://dx.doi.org/10.1016/j.seizure.2016.11.029.

[61] Martijn J. Oude Wolcherink et al. “Health Economic Research Assessing the Value of Early Detection of Cardiovascular Disease: A Systematic Review”. In: *PharmacoEconomics* 41.10 (June 2023), pp. 1183–1203. ISSN: 1179-2027. DOI: 10.1007/s40273-023-01287-2. URL: http://dx.doi.org/10.1007/s40273-023-01287-2.

[62] Ahmet Remzi Ozcan and Sarp Erturk. “Seizure Prediction in Scalp EEG Using 3D Convolutional Neural Networks With an Image-Based Approach”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27.11 (Nov. 2019), pp. 2284–2293. ISSN: 1558-0210. DOI: 10.1109/tnsre.2019.2943707. URL: http://dx.doi.org/10.1109/TNSRE.2019.2943707.

[63] Eric T. Payne et al. “Seizure burden is independently associated with short term outcome in critically ill children”. In: *Brain* 137.5 (Mar. 2014), pp. 1429–1438. ISSN: 0006-8950. DOI: 10.1093/brain/awu042. URL: http://dx.doi.org/10.1093/brain/awu042.

[64] Gyrithe L. Pedersen et al. “Prognostic value of periodic electroencephalographic discharges for neurological patients with profound disturbances of consciousness”. In: *Clinical Neurophysiology* 124.1 (Jan. 2013), pp. 44–51. ISSN: 1388-2457. DOI: 10.1016/j.clinph.2012.06.010. URL: http://dx.doi.org/10.1016/j.clinph.2012.06.010.

[65] Mangor Pedersen et al. “Artificial intelligence for clinical decision support in neurology”. In: *Brain Communications* 2.2 (2020). ISSN: 2632-1297. DOI: 10.1093/braincomms/fcaa096. URL: http://dx.doi.org/10.1093/braincomms/fcaa096.

[66] Dinh Q. Phung et al. “Using Shannon Entropy as EEG Signal Feature for Fast Person Identification”. In: *The European Symposium on Artificial Neural Networks*. 2014. URL: https://api.semanticscholar.org/CorpusID:13910513.

[67] Diana Gina Poalelungi et al. “Advancing Patient Care: How Artificial Intelligence Is Transforming Healthcare”. In: *Journal of Personalized Medicine* 13.8 (July 2023), p. 1214. ISSN: 2075-4426. DOI: 10.3390/jpm13081214. URL: http://dx.doi.org/10.3390/jpm13081214.

[68] Jan Pyrzowski et al. “Zero-crossing patterns reveal subtle epileptiform discharges in the scalp EEG”. In: *Scientific Reports* 11.1 (Feb. 2021). ISSN: 2045-2322. DOI: 10.1038/s41598-021-83337-3. URL: http://dx.doi.org/10.1038/s41598-021-83337-3.

[69] Andres Rodriguez Ruiz et al. “Association of Periodic and Rhythmic Electroencephalographic Patterns With Seizures in Critically Ill Patients”. In: *JAMA Neurology* 74.2 (Feb. 2017), p. 181. ISSN: 2168-6149. DOI: 10.1001/jamaneurol.2016.4990. URL: http://dx.doi.org/10.1001/jamaneurol.2016.4990.

[70] Siming Rong et al. “Abnormal Neural Activity in Different Frequency Bands in Parkinson’s Disease With Mild Cognitive Impairment”. In: *Frontiers in Aging Neuroscience* 13 (Aug. 2021). ISSN: 1663-4365. DOI: 10.3389/fnagi.2021.709998. URL: http://dx.doi.org/10.3389/fnagi.2021.709998.

[71] Michael A. Rosen et al. “Teamwork in healthcare: Key discoveries enabling safer, high-quality care.” In: *American Psychologist* 73.4 (May 2018), pp. 433–450. ISSN: 0003-066X. DOI: 10.1037/amp0000298. URL: http://dx.doi.org/10.1037/amp0000298.

[72] Ahmad Waleed Salehi et al. “A Study of CNN and Transfer Learning in Medical Imaging: Advantages, Challenges, Future Scope”. In: *Sustainability* 15.7 (Mar. 2023), p. 5930. ISSN: 2071-1050. DOI: 10.3390/su15075930. URL: http://dx.doi.org/10.3390/su15075930.

[73] Sani Saminu et al. “Applications of Artificial Intelligence in Automatic Detection of Epileptic Seizures Using EEG Signals: A Review”. In: *Artificial Intelligence and Applications* 1.1 (Aug. 2022), pp. 11–25. ISSN: 2811-0854. DOI: 10.47852/bonviewaia2202297. URL: http://dx.doi.org/10.47852/bonviewAIA2202297.

[74] Sarah E. Schmitt. “Generalized and Lateralized Rhythmic Patterns”. In: *Journal of Clinical Neurophysiology* 35.3 (May 2018), pp. 218–228. ISSN: 0736-0258. DOI: 10.1097/wnp.0000000000000446. URL: http://dx.doi.org/10.1097/wnp.0000000000000446.

[75] Adithya Sivaraju and Emily J. Gilmore. “Understanding and Managing the Ictal-Interictal Continuum in Neurocritical Care”. In: *Current Treatment Options in Neurology* 18.2 (Feb. 2016). ISSN: 1534-3138. DOI: 10.1007/s11940-015-0391-0. URL: http://dx.doi.org/10.1007/s11940-015-0391-0.

[76] Itaf Ben Slimen et al. “EEG epileptic seizure detection and classification based on dual-tree complex wavelet transform and machine learning algorithms”. In: *The Journal of Biomedical Research* 34.3 (2020), p. 151. ISSN: 1674-8301. DOI: 10.7555/jbr.34.20190026. URL: http://dx.doi.org/10.7555/JBR.34.20190026.

[77] S J M Smith. “EEG in the diagnosis, classification, and management of patients with epilepsy”. In: *Journal of Neurology, Neurosurgery amp; Psychiatry* 76.suppl$_2$ (June 2005), pp. ii2–ii7. ISSN: 0022-3050. DOI: 10.1136/jnnp.2005.069245. URL: http://dx.doi.org/10.1136/jnnp.2005.069245.

[78] D. Puthankattil Subha et al. “EEG Signal Analysis: A Survey”. In: *Journal of Medical Systems* 34.2 (Dec. 2008), pp. 195–212. ISSN: 1573-689X. DOI: 10.1007/s10916-008-9231-z. URL: http://dx.doi.org/10.1007/s10916-008-9231-z.

[79] Yuna Sugianela, Qonita Luthfia Sutino, and Darlis Herumurti. “EEG CLASSIFICATION FOR EPILEPSY BASED ON WAVELET PACKET DECOMPOSITION AND RANDOM FOREST”. In: *Jurnal Ilmu Komputer dan Informasi* 11.1 (Feb. 2018), p. 27. ISSN: 2088-7051. DOI: 10.21609/jiki.v11i1.549. URL: http://dx.doi.org/10.21609/jiki.v11i1.549.

[80] Christa B. Swisher et al. “Baseline EEG Pattern on Continuous ICU EEG Monitoring and Incidence of Seizures”. In: *Journal of Clinical Neurophysiology* 32.2 (Apr. 2015), pp. 147–151. ISSN: 0736-0258. DOI: 10.1097/wnp.0000000000000157. URL: http://dx.doi.org/10.1097/wnp.0000000000000157.

[81] James X. Tao, Xiaoxiao Qin, and Qun Wang. “Ictal-interictal continuum: a review of recent advancements”. In: *Acta Epileptologica* 2.1 (Aug. 2020). ISSN: 2524-4434. DOI: 10.1186/s42494-020-00021-1. URL: http://dx.doi.org/10.1186/s42494-020-00021-1.

[82] Kostas M. Tsiouris et al. “A Long Short-Term Memory deep learning network for the prediction of epileptic seizures using EEG signals”. In: *Computers in Biology and Medicine* 99 (Aug. 2018), pp. 24–37. ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2018.05.019. URL: http://dx.doi.org/10.1016/j.compbiomed.2018.05.019.

[83] Shafiq Ul Rehman et al. “AI-based tool for early detection of Alzheimer’s disease”. In: *Heliyon* 10.8 (Apr. 2024), e29375. ISSN: 2405-8440. DOI: 10.1016/j.heliyon.2024.e29375. URL: http://dx.doi.org/10.1016/j.heliyon.2024.e29375.

[84] Acharya UR et al. “Automated diagnosis of epileptic electroencephalogram using independent component analysis and discrete wavelet transform for different electroencephalogram durations”. In: *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine* 227.3 (Dec. 2012), pp. 234–244. ISSN: 2041-3033. DOI: 10.1177/0954411912467883. URL: http://dx.doi.org/10.1177/0954411912467883.

[85] Paul Vanabelle et al. “Epileptic seizure detection using EEG signals and extreme gradient boosting”. In: *The Journal of Biomedical Research* 34.3 (2020), p. 228. ISSN: 1674-8301. DOI: 10.7555/jbr.33.20190016. URL: http://dx.doi.org/10.7555/JBR.33.20190016.

[86] Bart Vanrumste et al. “Slow-wave activity arising from the same area as epileptiform activity in the EEG of paediatric patients with focal epilepsy”. In: *Clinical Neurophysiology* 116.1 (Jan. 2005), pp. 9–17. ISSN: 1388-2457. DOI: 10.1016/j.clinph.2004.07.032. URL: http://dx.doi.org/10.1016/j.clinph.2004.07.032.

[87] Jin Wang et al. “Bag-of-words representation for biomedical time series classification”. In: *Biomedical Signal Processing and Control* 8.6 (Nov. 2013), pp. 634–644. ISSN: 1746-8094. DOI: 10.1016/j.bspc.2013.06.004. URL: http://dx.doi.org/10.1016/j.bspc.2013.06.004.

[88] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. “A survey of transfer learning”. In: *Journal of Big Data* 3.1 (May 2016). ISSN: 2196-1115. DOI: 10.1186/s40537-016-0043-6. URL: http://dx.doi.org/10.1186/s40537-016-0043-6.

[89] Tina Willmen et al. “Health economic benefits through the use of diagnostic support systems and expert knowledge”. In: *BMC Health Services Research* 21.1 (Sept. 2021). ISSN: 1472-6963. DOI: 10.1186/s12913-021-06926-y. URL: http://dx.doi.org/10.1186/s12913-021-06926-y.

[90] Sheng Wong et al. “<scp>EEG</scp> datasets for seizure detection and prediction— A review”. In: *Epilepsia Open* 8.2 (Feb. 2023), pp. 252–267. ISSN: 2470-9239. DOI: 10.1002/epi4.12704. URL: http://dx.doi.org/10.1002/epi4.12704.

[91] Jennifer Wu et al. “Utility of EEG measures of brain function in patients with acute stroke”. en. In: *J. Neurophysiol.* 115.5 (May 2016), pp. 2399–2405.

[92] Tiantian Xiao et al. “Self-supervised Learning with Attention Mechanism for EEG-based seizure detection”. In: *Biomedical Signal Processing and Control* 87 (Jan. 2024), p. 105464. ISSN: 1746-8094. DOI: 10.1016/j.bspc.2023.105464. URL: http://dx.doi.org/10.1016/j.bspc.2023.105464.

[93] Zhen Xu et al. *Learning an Adaptive Learning Rate Schedule*. 2019. DOI: 10.48550/ARXIV.1909.09712. URL: https://arxiv.org/abs/1909.09712.

[94] Peter Z. Yan et al. “Automated spectrographic seizure detection using convolutional neural networks”. In: *Seizure* 71 (Oct. 2019), pp. 124–131. ISSN: 1059-1311. DOI: 10.1016/j.seizure.2019.07.009. URL: http://dx.doi.org/10.1016/j.seizure.2019.07.009.

[95] Zuyi Yu et al. “Epileptic seizure prediction based on local mean decomposition and deep convolutional neural network”. In: *The Journal of Supercomputing* 76.5 (Sept. 2018), pp. 3462–3476. ISSN: 1573-0484. DOI: 10.1007/s11227-018-2600-6. URL: http://dx.doi.org/10.1007/s11227-018-2600-6.

[96] Sahar Zafar et al. “36: BURDEN OF EEG ICTAL-INTERICTAL CONTINUUM ACTIVITY PREDICTS POOR OUTCOME IN CRITICALLY ILL PATIENTS”. In: *Critical Care Medicine* 48.1 (Jan. 2020), pp. 18–18. ISSN: 0090-3493. DOI: 10.1097/01.ccm.0000618644.40877.19. URL: http://dx.doi.org/10.1097/01.ccm.0000618644.40877.19.

[97] Baocan Zhang et al. “Cross-Subject Seizure Detection in EEGs Using Deep Transfer Learning”. In: *Computational and Mathematical Methods in Medicine* 2020 (May 2020), pp. 1–8. ISSN: 1748-6718. DOI: 10.1155/2020/7902072. URL: http://dx.doi.org/10.1155/2020/7902072.

[98] Wei-Long Zheng et al. “Automated EEG-based prediction of delayed cerebral ischemia after subarachnoid hemorrhage”. In: *Clinical Neurophysiology* 143 (Nov. 2022), pp. 97–106. ISSN: 1388-2457. DOI: 10.1016/j.clinph.2022.08.023. URL: http://dx.doi.org/10.1016/j.clinph.2022.08.023.

[99] Laura Zwaan and Hardeep Singh. “The challenges in defining and measuring diagnostic error”. In: *Diagnosis* 2.2 (Mar. 2015), pp. 97–103. ISSN: 2194-8011. DOI: 10.1515/dx-2014-0069. URL: http://dx.doi.org/10.1515/dx-2014-0069.

A

Supplementary Materials

A.1. Bag of Words and Change Point detection parameters

subject_id	num_words	num_clusters	sgolay_para	thr_cpd	flag
glad003	50	50	10	0.1	3
glad032	50	30	10	0.1	1
glad046	100	50	10	0.1	1
glad027	100	50	10	0.1	1
glad018	50	30	10	0.1	4
glad023	100	50	10	0.1	4
glad094	50	30	10	0.1	1
glad078	50	30	10	0.1	1
glad059	50	30	10	0.1	5
glad131	50	30	10	0.1	1
goodSZ001	50	30	10	0.1	1
goodSZ002	50	30	10	0.1	1
goodSZ004	50	30	10	0.1	1
goodSZ005	50	30	10	0.1	1
goodSZ006	100	50	10	0.1	1
glad053	50	30	10	0.1	5
glad007	50	30	10	0.1	5
glad057	50	30	10	0.1	1
glad060	50	30	10	0.1	1
glad125	50	30	10	0.1	1
glad062	50	50	10	0.1	1
glad109	50	30	10	0.1	3
glad102	50	30	10	0.1	2
glad115	100	50	10	0.1	2
glad127	50	30	10	0.1	2
glad013	100	50	10	0.1	1
glad022	100	50	10	0.1	1
glad001	50	30	10	0.1	0
glad002	50	30	10	0.1	1
glad004	50	30	10	0.1	1
glad005	50	30	10	0.1	1
glad006	50	30	10	0.1	1
glad008	50	30	10	0.1	1
glad009	50	30	10	0.1	1
glad010	50	30	10	0.1	1
glad011	100	50	10	0.1	1

Table Continued from previous page

subject_id	num_words	num_clusters	sgolay_para	thr_cpd	flag
glad012	100	50	10	0.1	1
glad014	100	50	10	0.1	1
glad015	50	30	10	0.1	1
glad016	100	50	10	0.1	1
glad017	100	50	10	0.1	1
glad019	50	30	10	0.1	1
glad020	50	30	10	0.1	1
glad021	50	30	10	0.1	1
glad024	50	30	10	0.1	1
glad025	50	30	10	0.1	1
glad026	50	30	10	0.1	1
glad028	50	30	10	0.1	1
glad029	50	30	10	0.1	1
glad030	50	30	10	0.1	1
glad031	50	30	10	0.1	1
glad033	50	30	10	0.1	1
glad034	50	30	10	0.1	1
glad035	50	30	10	0.1	1
glad036	50	30	10	0.1	1
glad037	50	30	10	0.1	1
glad038	100	50	10	0.1	1
glad039	50	30	10	0.1	1
glad040	50	30	10	0.1	1
glad041	50	30	10	0.1	1
glad042	50	30	10	0.1	1
glad043	50	30	10	0.1	1
glad044	50	30	10	0.1	1
glad045	50	30	10	0.1	1
glad047	100	50	10	0.1	1
glad048	50	30	10	0.1	1
glad049	50	30	10	0.1	1
glad050	50	30	10	0.1	1
glad051	50	30	10	0.1	1
glad052	50	30	10	0.1	1
glad054	50	30	10	0.1	1
glad055	50	30	10	0.1	1
glad056	50	30	10	0.1	1
glad058	50	30	10	0.1	1
glad061	50	30	10	0.1	1
glad063	50	50	10	0.1	1
glad064	50	30	10	0.1	1
glad065	50	30	10	0.1	1
glad066	50	30	10	0.1	1
glad067	50	30	10	0.1	1
glad068	50	30	10	0.1	1
glad069	50	30	10	0.1	1
glad070	50	30	10	0.1	1
glad071	50	30	10	0.1	1
glad073	50	30	10	0.1	1
glad074	50	30	10	0.1	1
glad075	100	50	10	0.1	1
glad076	50	30	10	0.1	1
glad077	50	30	10	0.1	1
glad079	50	30	10	0.1	1

Table Continued from previous page

subject_id	num_words	num_clusters	sgolay_para	thr_cpd	flag
glad080	50	30	10	0.1	1
glad081	50	30	10	0.1	1
glad082	50	30	10	0.1	1
glad083	50	30	10	0.1	1
glad084	50	30	10	0.1	1
glad085	50	30	10	0.1	1
glad086	50	30	10	0.1	1
glad087	50	30	10	0.1	1
glad088	50	30	10	0.1	1
glad089	50	30	10	0.1	1
glad090	50	30	10	0.1	1
glad091	50	30	10	0.1	1
glad092	50	30	10	0.1	1
glad093	50	30	10	0.1	1
glad095	50	30	10	0.1	1
glad096	50	30	10	0.1	1
glad097	50	30	10	0.1	1
glad098	50	30	10	0.1	1
glad099	50	30	10	0.1	1
glad100	50	30	10	0.1	1
glad101	50	30	10	0.1	1
glad103	50	30	10	0.1	1
glad104	50	30	10	0.1	1
glad105	50	30	10	0.1	1
glad106	50	30	10	0.1	1
glad107	50	30	10	0.1	1
glad108	50	30	10	0.1	1
glad110	50	30	10	0.1	1
glad112	50	30	10	0.1	1
glad113	50	30	10	0.1	1
glad114	50	30	10	0.1	1
glad116	50	30	10	0.1	1
glad117	50	30	10	0.1	1
glad118	50	30	10	0.1	1
glad119	50	30	10	0.1	1
glad120	50	30	10	0.1	1
glad121	50	30	10	0.1	1
glad122	50	30	10	0.1	1
glad123	50	30	10	0.1	1
glad124	50	30	10	0.1	1
glad126	50	30	10	0.1	1
glad128	50	30	10	0.1	1
glad129	50	30	10	0.1	1
glad130	50	30	10	0.1	1
glad132	50	30	10	0.1	1
glad133	50	30	10	0.1	1
glad134	50	30	10	0.1	1
glad135	50	30	10	0.1	1
glad158	50	30	10	0.1	1
glad159	50	30	10	0.1	1

A.2. Patient information

Patient	Gender	Age (years)	EEG Length (hours)
glad001	Female	58.6	15.7
glad002	Female	7.8	12.0
glad003	Male	56.0	12.8
glad004	Male	76.5	12.9
glad005	Male	9.7	12.0
glad006	Male	74.4	12.8
glad007	Male	33.7	13.0
glad008	Female	7.6	12.0
glad009	Female	87.3	12.7
glad010	Female	65.6	12.5
glad011	Male	67.8	12.7
glad012	Male	22.8	12.0
glad013	Male	77.8	12.4
glad014	Male	59.0	13.3
glad015	Male	76.5	12.7
glad016	Male	18.2	12.0
glad017	Female	62.3	12.8
glad018	Male	11.4	16.2
glad019	Male	59.3	15.7
glad020	Female	22.0	12.7
glad021	Male	39.5	15.5
glad022	Female	35.6	12.8
glad023	Male	68.6	23.2
glad024	Female	67.3	13.2
glad025	Female	57.7	12.0
glad026	Male	57.7	14.6
glad027	Male	57.9	12.4
glad028	Female	58.4	13.0
glad029	Male	60.1	12.5
glad030	Male	43.3	12.0
glad031	Female	74.0	12.4
glad032	Male	3.4	12.0
glad033	Male	76.0	20.3
glad034	Female	54.8	12.2
glad035	Female	48.8	18.4
glad036	Female	12.3	12.0
glad037	Female	74.4	12.5
glad038	Male	16.8	12.0
glad039	Female	78.9	17.8
glad040	Female	67.8	12.0
glad041	Male	63.4	13.2
glad042	Male	63.0	12.7
glad043	Male	83.6	12.8
glad044	Female	64.8	12.6
glad045	Male	0.0	12.0
glad046	Male	61.3	13.8
glad047	Female	42.7	12.9
glad048	Female	25.4	12.0
glad049	Female	67.0	12.5
glad050	Female	79.1	13.0
glad051	Female	29.0	12.0
glad052	Male	84.7	12.6
glad053	Male	20.1	12.0
glad054	Male	62.3	16.7

glad055	Female	83.7	12.8
glad056	Female	26.5	12.0
glad057	Male	59.2	13.0
glad058	Male	7.4	12.0
glad059	Male	28.7	12.0
glad060	Male	37.6	12.8
glad061	Female	57.4	12.0
glad062	Female	86.2	12.7
glad063	Female	64.6	12.0
glad064	Male	83.3	12.0
glad065	Male	45.9	12.4
glad066	Female	32.0	12.0
glad067	Female	5.3	13.2
glad068	Male	7.7	12.0
glad069	Female	64.9	12.7
glad070	Female	74.6	12.8
glad071	Female	38.7	16.0
glad073	Female	84.3	12.9
glad074	Male	71.7	12.5
glad075	Female	10.0	12.0
glad076	Female	21.1	12.0
glad077	Male	19.2	22.0
glad078	Male	30.3	14.8
glad079	Female	49.1	18.9
glad080	Male	63.6	12.4
glad081	Male	13.9	12.0
glad082	Male	26.3	12.0
glad083	Female	6.8	12.0
glad084	Female	61.4	12.4
glad085	Female	20.6	12.0
glad086	Male	40.3	12.0
glad087	Female	49.6	12.0
glad088	Female	47.8	12.8
glad089	Male	17.9	12.7
glad090	Male	26.0	14.9
glad091	Male	1.9	12.0
glad092	Male	71.4	12.9
glad093	Male	73.1	14.7
glad094	Male	51.9	13.2
glad095	Male	61.0	13.5
glad096	Female	23.8	12.0
glad097	Female	71.5	12.6
glad098	Male	47.8	12.0
glad099	Female	71.2	12.6
glad100	Female	64.3	20.8
glad101	Male	82.9	12.0
glad102	Female	86.5	15.4
glad103	Female	20.8	12.0
glad104	Female	72.7	12.6
glad105	Male	59.3	12.6
glad106	Female	57.0	16.1
glad107	Female	69.9	12.0
glad108	Female	69.1	12.7
glad109	Female	68.3	13.2
glad110	Male	79.2	18.0
glad112	Male	64.0	13.0

glad113	Female	79.4	12.0
glad114	Female	77.5	12.6
glad115	Female	64.8	13.1
glad116	Female	47.2	12.0
glad117	Female	79.9	20.5
glad118	Female	2.6	13.3
glad119	Male	50.1	12.0
glad120	Male	89.2	13.1
glad121	Female	53.9	13.2
glad122	Female	74.9	12.9
glad123	Female	40.1	12.0
glad124	Male	72.8	14.4
glad125	Male	36.5	12.0
glad126	Male	13.0	12.0
glad127	Male	26.7	12.8
glad128	Female	20.6	12.0
glad129	Male	19.7	12.0
glad130	Male	26.1	12.0
glad131	Male	72.5	12.6
glad132	Male	33.3	12.0
glad133	Female	65.3	15.6
glad134	Male	56.1	12.0
glad135	Female	68.8	12.0
glad158	Female	57.3	12.0
glad159	Male	83.4	21.2
flame01	Female	87.8	12.0
flame02	Female	87.5	12.0
flame04	Male	76.5	12.0
flame05	Female	54.2	12.0
flame06	Male	20.8	12.0

A.3. DenseNet convolutional neural network architecture

Layer (type)	Output Shape	Param #
Conv1d-1	[-1, 64, 1000]	7,232
ELU-2	[-1, 64, 1000]	0
MaxPool1d-3	[-1, 64, 500]	0
ELU-4	[-1, 64, 500]	0
Conv1d-5	[-1, 128, 500]	8,320
ELU-6	[-1, 128, 500]	0
Conv1d-7	[-1, 32, 500]	12,320
ELU-8	[-1, 96, 500]	0
Conv1d-9	[-1, 128, 500]	12,416
ELU-10	[-1, 128, 500]	0
Conv1d-11	[-1, 32, 500]	12,320
ELU-12	[-1, 128, 500]	0
Conv1d-13	[-1, 128, 500]	16,512
ELU-14	[-1, 128, 500]	0
Conv1d-15	[-1, 32, 500]	12,320
ELU-16	[-1, 160, 500]	0
Conv1d-17	[-1, 128, 500]	20,608
ELU-18	[-1, 128, 500]	0
Conv1d-19	[-1, 32, 500]	12,320
ELU-20	[-1, 192, 500]	0
Conv1d-21	[-1, 96, 500]	18,528

AvgPool1d-22	[-1, 96, 250]	0
ELU-23	[-1, 96, 250]	0
Conv1d-24	[-1, 128, 250]	12,416
ELU-25	[-1, 128, 250]	0
Conv1d-26	[-1, 32, 250]	12,320
ELU-27	[-1, 128, 250]	0
Conv1d-28	[-1, 128, 250]	16,512
ELU-29	[-1, 128, 250]	0
Conv1d-30	[-1, 32, 250]	12,320
ELU-31	[-1, 160, 250]	0
Conv1d-32	[-1, 128, 250]	20,608
ELU-33	[-1, 128, 250]	0
Conv1d-34	[-1, 32, 250]	12,320
ELU-35	[-1, 192, 250]	0
Conv1d-36	[-1, 128, 250]	24,704
ELU-37	[-1, 128, 250]	0
Conv1d-38	[-1, 32, 250]	12,320
ELU-39	[-1, 224, 250]	0
Conv1d-40	[-1, 112, 250]	25,200
AvgPool1d-41	[-1, 112, 125]	0
ELU-42	[-1, 112, 125]	0
Conv1d-43	[-1, 128, 125]	14,464
ELU-44	[-1, 128, 125]	0
Conv1d-45	[-1, 32, 125]	12,320
ELU-46	[-1, 144, 125]	0
Conv1d-47	[-1, 128, 125]	18,560
ELU-48	[-1, 128, 125]	0
Conv1d-49	[-1, 32, 125]	12,320
ELU-50	[-1, 176, 125]	0
Conv1d-51	[-1, 128, 125]	22,656
ELU-52	[-1, 128, 125]	0
Conv1d-53	[-1, 32, 125]	12,320
ELU-54	[-1, 208, 125]	0
Conv1d-55	[-1, 128, 125]	26,752
ELU-56	[-1, 128, 125]	0
Conv1d-57	[-1, 32, 125]	12,320
ELU-58	[-1, 240, 125]	0
Conv1d-59	[-1, 120, 125]	28,920
AvgPool1d-60	[-1, 120, 62]	0
ELU-61	[-1, 120, 62]	0
Conv1d-62	[-1, 128, 62]	15,488
ELU-63	[-1, 128, 62]	0
Conv1d-64	[-1, 32, 62]	12,320
ELU-65	[-1, 152, 62]	0
Conv1d-66	[-1, 128, 62]	19,584
ELU-67	[-1, 128, 62]	0
Conv1d-68	[-1, 32, 62]	12,320
ELU-69	[-1, 184, 62]	0
Conv1d-70	[-1, 128, 62]	23,680
ELU-71	[-1, 128, 62]	0
Conv1d-72	[-1, 32, 62]	12,320
ELU-73	[-1, 216, 62]	0
Conv1d-74	[-1, 128, 62]	27,776
ELU-75	[-1, 128, 62]	0
Conv1d-76	[-1, 32, 62]	12,320
ELU-77	[-1, 248, 62]	0

Conv1d-78	[-1, 124, 62]	30,876
AvgPool1d-79	[-1, 124, 31]	0
ELU-80	[-1, 124, 31]	0
Conv1d-81	[-1, 128, 31]	16,000
ELU-82	[-1, 128, 31]	0
Conv1d-83	[-1, 32, 31]	12,320
ELU-84	[-1, 156, 31]	0
Conv1d-85	[-1, 128, 31]	20,096
ELU-86	[-1, 128, 31]	0
Conv1d-87	[-1, 32, 31]	12,320
ELU-88	[-1, 188, 31]	0
Conv1d-89	[-1, 128, 31]	24,192
ELU-90	[-1, 128, 31]	0
Conv1d-91	[-1, 32, 31]	12,320
ELU-92	[-1, 220, 31]	0
Conv1d-93	[-1, 128, 31]	28,288
ELU-94	[-1, 128, 31]	0
Conv1d-95	[-1, 32, 31]	12,320
ELU-96	[-1, 252, 31]	0
Conv1d-97	[-1, 126, 31]	31,878
AvgPool1d-98	[-1, 126, 15]	0
ELU-99	[-1, 126, 15]	0
Conv1d-100	[-1, 128, 15]	16,256
ELU-101	[-1, 128, 15]	0
Conv1d-102	[-1, 32, 15]	12,320
ELU-103	[-1, 158, 15]	0
Conv1d-104	[-1, 128, 15]	20,352
ELU-105	[-1, 128, 15]	0
Conv1d-106	[-1, 32, 15]	12,320
ELU-107	[-1, 190, 15]	0
Conv1d-108	[-1, 128, 15]	24,448
ELU-109	[-1, 128, 15]	0
Conv1d-110	[-1, 32, 15]	12,320
ELU-111	[-1, 222, 15]	0
Conv1d-112	[-1, 128, 15]	28,544
ELU-113	[-1, 128, 15]	0
Conv1d-114	[-1, 32, 15]	12,320
ELU-115	[-1, 254, 15]	0
Conv1d-116	[-1, 127, 15]	32,385
AvgPool1d-117	[-1, 127, 7]	0
ELU-118	[-1, 127, 7]	0
Conv1d-119	[-1, 128, 7]	16,384
ELU-120	[-1, 128, 7]	0
Conv1d-121	[-1, 32, 7]	12,320
ELU-122	[-1, 159, 7]	0
Conv1d-123	[-1, 128, 7]	20,480
ELU-124	[-1, 128, 7]	0
Conv1d-125	[-1, 32, 7]	12,320
ELU-126	[-1, 191, 7]	0
Conv1d-127	[-1, 128, 7]	24,576
ELU-128	[-1, 128, 7]	0
Conv1d-129	[-1, 32, 7]	12,320
ELU-130	[-1, 223, 7]	0
Conv1d-131	[-1, 128, 7]	28,672
ELU-132	[-1, 128, 7]	0
Conv1d-133	[-1, 32, 7]	12,320

ReLU-134	[-1, 255, 7]	0
AvgPool1d-135	[-1, 255, 1]	0
DenseNetEnconder-136	[-1, 255]	0
Dropout-137	[-1, 255]	0
Linear-138	[-1, 6]	1,536
Total params:	1,090,859	
Trainable params:	1,090,859	
Non-trainable params:	0	

A.4. Training losses

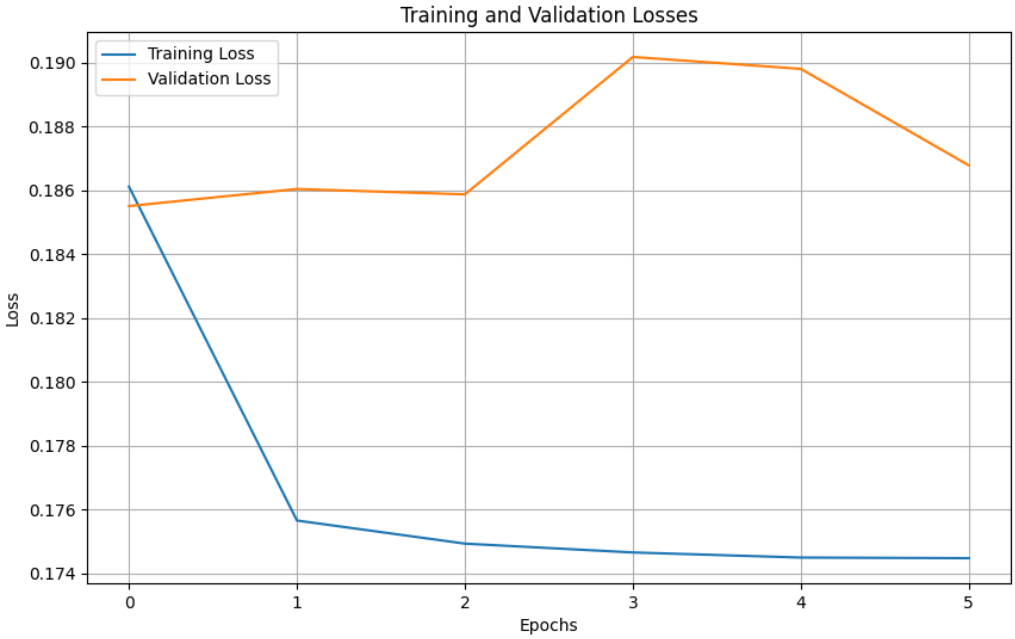


Figure A.1: Training loss: batch size 128, shuffle

Figure A.2: Training loss: batch size 256, denseblock 7

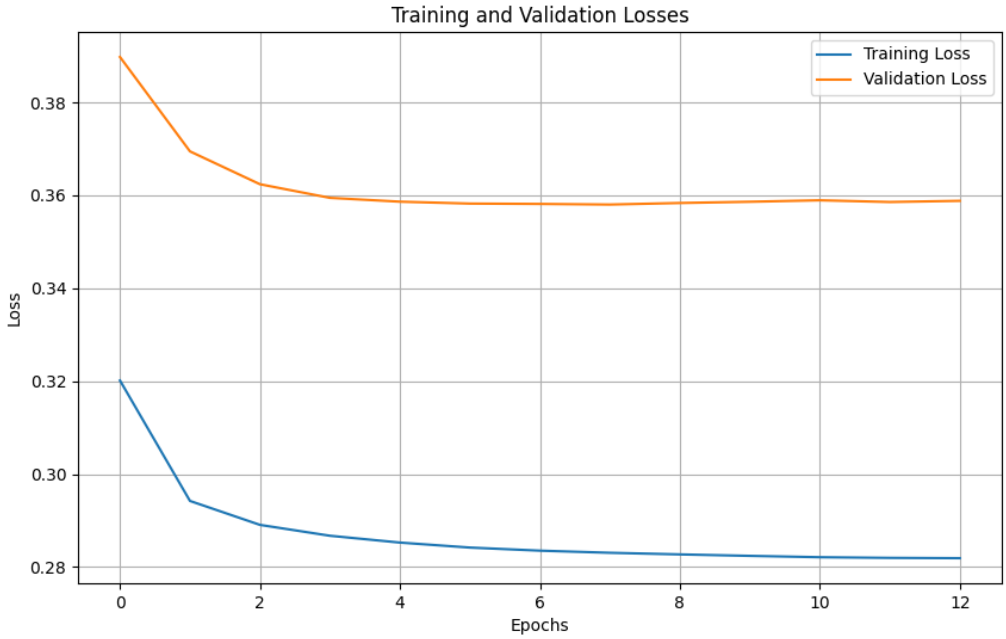


Figure A.3: Training loss: batch size 256

Figure A.4: Training loss: batch size 128, learning rate 0.1

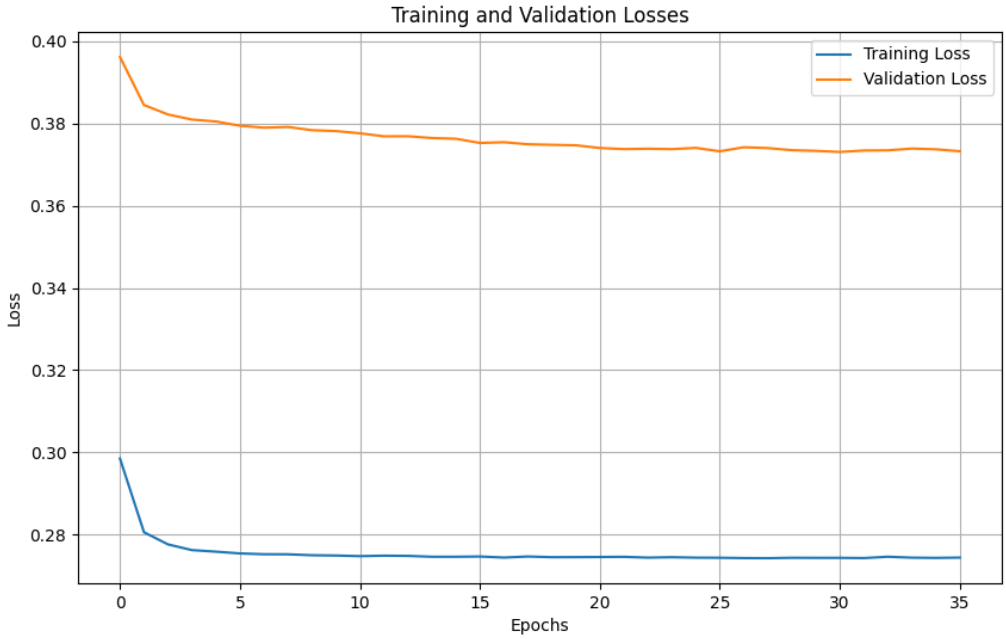


Figure A.5: Training loss: batch size 128

Figure A.6: Training loss: batch size 256, no transfer learning

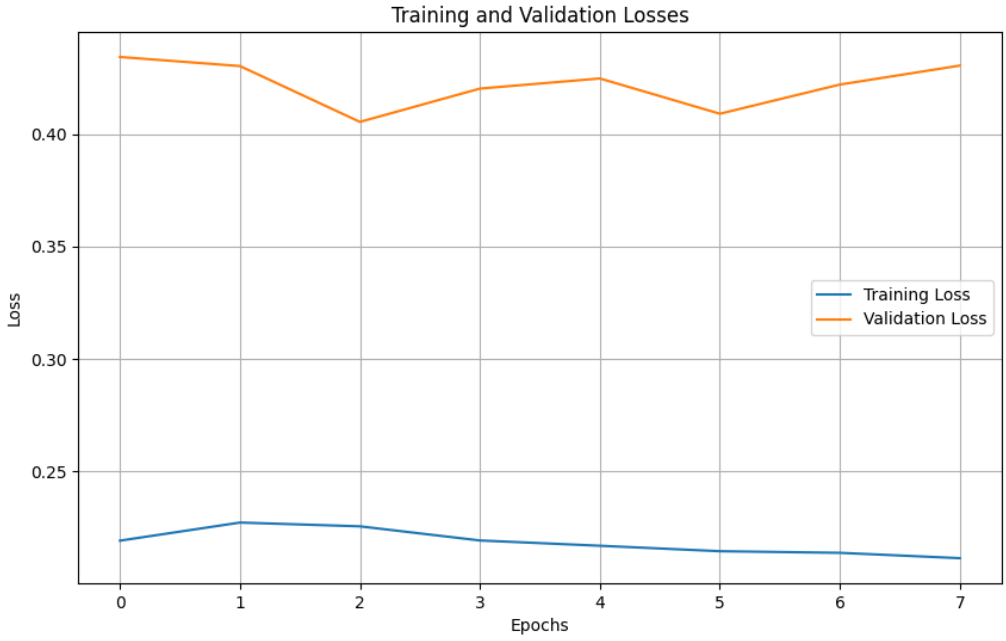


Figure A.7: Training loss: batch size 256, full model trained